

3  
Z MIKROKOMPUTEREM NA TY

NR INDEKSU 353965  
PL ISSN 0860-1674

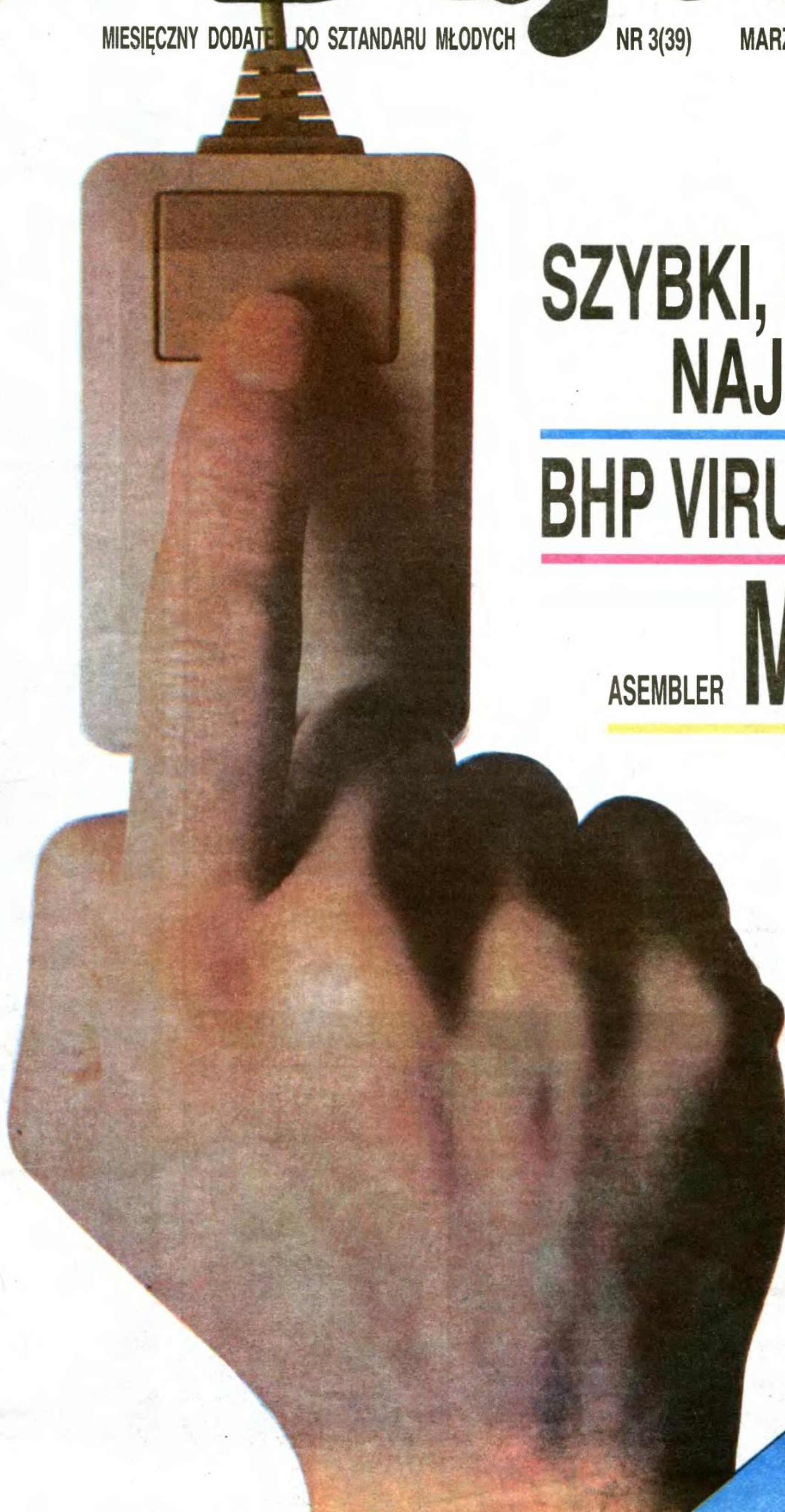
# Bajtek

MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH

NR 3(39)

MARZEC 1989

CENA 220 ZŁ



**SZYBKI, SZYBSZY  
NAJSZYBSZY**

**BHP VIRUS KILLER**

ASEMBLER

**MAC/65**

**TEST  
PLOTERA**

**KAL-  
KU-  
LA-  
TO-  
RY**

**KINGS' QUEST**

**SM**  
SZTANDAR  
MŁODYCH



# POZA PRIORYTETEM

W kilku poprzednich numerach pisałem w tym miejscu o najnowszych osiągnięciach technicznych i trendach rozwojowych informatyki na świecie. A co z nami? Gdzie jest Polska — w chwili, gdy inni zaczynają już realizować projekty rozwojowe obliczone na XXI wiek?

Otóż, niestety, zostaliśmy bardzo daleko z tyłu i wiele wskazuje na to, że niewiele jest chętnych do zmiany tej sytuacji. Elektronika nie zmieściła się bowiem wśród priorytetów rozwojowych kraju (za takowe uznano tylko rolnictwo, budownictwo i ochronę środowiska) i nie bardzo widać ludzi w ekipie rządzącej, którzy by rozumieli znaczenie elektroniki dla rozwoju współczesnego świata oraz chcieli się na jej rzecz zaangażować. Nie mamy swojego Jewgienija Wielichowa wśród grona czołowych doradców i będziemy płacić za to w najbliższych latach coraz większą cenę!

Skąd to zaniepokojenie — może ktoś zripostować — przecież mamy coraz więcej komputerów, których zazdroszczą nam sąsiedzi? Obawiam się, że to te właśnie komputery służą wielu ludziom za usprawiedliwienie niedoinwestowania elektroniki. No bo skoro stają już te cacka w wielu gabinetach i migają swymi kolorowymi ekranami, to przecież jest chyba dobrze, no nie?

Nie chciałbym raz jeszcze szeroko analizować fenomenu polskiego wariantu komputeryzacji, bo nieraz się już tym zajmowaliśmy. Stwierdzę tylko, że bardziej jest to fenomen z zakresu edukacji społecznej, niż rewolucji technologicznej w polskim przemyśle. Owszem, te, wszystkie ATARI, TIMEXY i AMSTRADY, a nawet montowane z dalekowschodnich klocków niby IBM-y wychowują nową generację Polaków, kształtują w niej nawyki życia w cywilizowanym społeczeństwie. Ale wpływ tej naszej informatyki od Sasa do Lasa na rzeczywisty wzrost społecznej wydajności pracy jest bliski zera. I jeśli to nowe „bajtkowe” (w odróżnieniu od „świerszczykowego”) pokolenie nie znajdzie możliwości wykorzystania w swym zawodowym życiu nabywanych aktualnie umiejętności, może to tylko powiększyć grono rozczarowanych, patrzących tęsknie za oceanem...

O wygranej w przyszłości zadecydują bowiem nie montowane z tajwańskich części klony, choć są one ważne a czasami nawet niezastąpione — tylko powszechne zastosowanie mikroprocesorów we wszystkich dziedzinach przemysłu. Żeby jednak te mikroprocesory stosować, trzeba je mieć. Obecnie potrzebujemy 100–200 mln układów scalonych rocznie. W 1995 roku będziemy ich potrzebować około 400–500 mln. Nie kupimy ich za eksport żywności, bo póki

co sami jemy dosyć skromnie. Za darmo też ich nie dostaniemy, bo tylko raz się w historii ludzkości zdarzyło, że coś spadło za darmo z nieba, i to akurat nie scalał... Trzeba je więc wyprodukować. I tu zaczynają się schody.

Na łamach „Przeglądu Tygodniowego” profesor Jarosław Świdorski polemizował niedawno z nadgorliwymi chwalcami priorytetów rządu twierdzącymi, że skoro pociąg z komputerami i robotami przemysłowymi odjechał parę lat temu, i skoro do niego wówczas nie wsiedliśmy, to nie ma co stawiać teraz na rozwój elektroniki.

Jest to założenie samobójcze — stwierdza Profesor. — Nasz kraj był opóźniony w dziedzinie elektryfikacji. Gdybyśmy nie postawili na upowszechnienie energii elektrycznej to w jakiej sytuacji byłibyśmy dzisiaj? Zdecydowaliśmy się jednak doganiać świat i dzięki temu obecnie jesteśmy na wyższym poziomie rozwoju. To samo należy odnieść do rozwoju mikroelektroniki. Bez niej żadna dziedzina nie będzie się szybko rozwijała. Ona decyduje o wydajności pracy. Pod koniec lat 70. mieliśmy mniej więcej taką samą wydajność pracy jak Czechosłowacja. Dzisiaj nasi sąsiedzi mają już dwukrotnie większą, ponieważ w znacznej mierze zautomatyzowali swój przemysł.

Najważniejsze było przed laty doprowadzenie do każdego użytkownika prądu elektrycznego — mówi Profesor. — Równie dostępne powinny być układy scalone, by Polak, tak jak Japończyk lub Amerykanin, mógł na podstawie katalogu zamówić te podzespoły, które są mu potrzebne do złożenia telewizora, komputera, maszyny do szycia. Jeśli nie zbudujemy odpowiedniej wytwórni, zostaniemy zepchnięci do poziomu krajów czwartego świata...

Dlaczego o tym wszystkim piszę? Z jednego, ale za to dosyć zasadniczego powodu: mam otóż wielką nadzieję, że to właśnie spośród Czytelników „Bajtki” wyrosną ludzie decydujący w przyszłości o rozwoju polskiego przemysłu. Jestem też przekonany, że tworzenie wielkiego, państwowego przemysłu elektronicznego z prawdziwego zdarzenia nie jest wcale mniej atrakcyjne od powoływania tak popularnych aktualnie prywatnych spółek typu „kupno-sprzedaj”. A żeby rozwijać w przyszłości polskie koncerny elektroniczne, trzeba się do tego przygotować już teraz.

Pomyślcie o tym drodzy Czytelnicy, w krótkiej przerwie podczas wczytywania przez Wasz komputer kolejnego programu.

Waldemar Siwiński

Już wkrótce — oprócz kolejnego, kwietniowego numeru „Bajtki”, w którym polecamy m.in. Atari Super Turbo, pierwsze kroki w assemblerze i fantastyczną grę komputerową, The Train — ukaże się na specjalne, dodatkowe wydanie „Bajtki — TYLKO DLA POCZĄTKUJĄCYCH”, a w nim:

- Wielka rodzina mikrokomputerów od Z (ZX81) do A (Amigi).
- Twój Komputer opowie ci o tym, jak masz się nim posługiwać.
- Co, gdzie i za ile? Zanim wybierzesz komputer, warto wiedzieć, ile trzeba za niego zapłacić.
- Gry komputerowe — dla każdego coś emocjonującego.
- Do czego służy komputer, czyli jak zostać uczniem czarnoksiężnika.
- Co jest w środku, czyli, czy druty potrafią myśleć?

Jeśli chcesz dowiedzieć się, co to są komputery, na jakich zasadach działają i do czego służą,

jeśli zamierzasz kupić komputer sobie, swojemu dziecku... albo swojej babci, a nie wiesz jaki typ wybrać (i za ile),

jeśli chcesz nauczyć się programować, a może nawet zostać w przyszłości informatykiem,

**NIE NAMYŚLAJ SIĘ ANI CHWILI! KUP „BAJTKA TYLKO DLA POCZĄTKUJĄCYCH” SOBIE, LUB MŁODSZEMU BRATU.**

**TATA DYREKTOR TEŻ UCIESZY SIĘ Z TAKIEGO PREZENTU PYTAJ W KIOSKU RUCHU PAMIĘTAJ: TA INWESTYCJA OPLACI CI SIĘ!**



# Komputer

**„BAJTEK” — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH”**

**ADRES:** 00-687 Warszawa, ul. Wspólna 61. Tel. 21-12-05 Przewodniczący Rady Redakcyjnej: Jerzy Domański — redaktor naczelny „Sztandaru Młodych”.

**ZESPÓŁ REDAKCYJNY:** Waldemar Siwiński (z-ca redaktora naczelnego „SM” — kierownik zespołu „Bajtki”), Grzegorz Onichimowski (sekretarz redakcji „Bajtki”), Roman Poznański (kierownik działu klanów), Krzysztof Czernek, Sławomir Gajda (red. techniczny), Andrzej Pilaszek, Sławomir Polak, Wanda Roszkowska (opr. graficzne), Kazimierz Treger, Marcin Waligórski, Roman Wojciechowski. Zdjęcia w numerze: Leopold Dzikowski

**kłany redagują:** Commodore — Klaudiusz Dybowski, Dominik Falkowski, Amstrad-Schneider — Jonasz Mayer, Spectrum — Marcin Przasnyski, Atari — Wojciech Zientara, Sergiusz Piotrowski

**Fotokład** — Tadeusz Olczak, **Montaż offsetowy** — Grażyna Ostaszewska, **Korekta** — Maria Krajewska, Zofia Wójtanska.

**WYDAWCA:** RSW „Prasa-Książka-Ruch” Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefon: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403, 414.

Cena 150 zł. Skład techniką CRT-200, przygotowania offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIAZKA-RUCH” w Ciechanowie, ul. Sienkiewicza 51. Nr zlecenia 12439 n. 150 000 egz. A-111





# ILE KOSZTUJE TWIERDZENIE PITAGORASA?

## OD EMAL-a DO...

**rozmowa  
z prof. Romualdem  
W. Marczyńskim,  
twórcą pierwszych,  
polskich  
komputerów**

— **Panie Profesorze, pańska przygoda z maszynami liczącymi zaczęła się bardzo dawno.**

— W roku 1946 byłem studentem Wydziału Elektrycznego Politechniki Warszawskiej. Wtedy to wpadł mi w ręce szósty numer miesięcznika „Problemy” zawierający informację o uruchomieniu w Stanach Zjednoczonych ENIAC-a.

Po raz pierwszy dowiedziałem się o istnieniu elektronicznych maszyn cyfrowych. Po ukończeniu studiów podjąłem pracę jako asystent w Katedrze Radiotechniki Politechniki Warszawskiej u prof. Groszkowskiego i prof. Ryżki. 23 grudnia tego roku w znajdującym się wtedy na ulicy Hożej Instytucie Fizyki doszło do „historycznego” jak się potem okazało, spotkania. Było nas sześciu — profesorowie Kazimierz Kuratowski i Andrzej Mostowski, doktor Henryk Greniowski i trzech inżynierów Krystyn Bochenek, Leon Łukaszewicz i ja. Zastanawialiśmy się nad możliwością budowy w kraju Aparatów Matematycznych — o słowie „komputer” nikt wtedy jeszcze nie słyszał. Profesor Kazimierz Kuratowski powiedział nam, że matematyka polska powinna zająć się maszynami liczącymi. Po utworzeniu w roku 1950 Instytutu Matematycznego przystąpiliśmy do organizowania Grupy Aparatów Matematycznych i zaczęliśmy pierwsze prace. Ja zająłem się maszynami cyfrowymi zaś Bochenek i Łukaszewicz maszynami analogowymi.

— **Czy mieliście kontakty z innymi ośrodkami prowadzącymi podobne badania?**

— Praktycznie do roku 1954 nie mieliśmy kontaktów z zagranicą, za wyjątkiem Czechosłowacji, gdzie Antonin Svoboda projektował swoją przekątnikową maszynę SAPO.

Była to pierwsza maszyna w świecie tolerująca

błędy. Nie miała ona jednak praktycznie wpływu na nasze projekty. Otrzymaliśmy od niego tylko pomoc w zakresie literatury.

— **Pierwsze polskie maszyny liczące powstały w latach pięćdziesiątych?**

— W okresie od 1952 roku do 1959 zbudowano cztery takie maszyny, były to EMAL (Elektryczna Maszyna Automatycznie Licząca), XYZ, EMAL — 2 i BINEG. W roku 1953 Leon Łukaszewicz ukończył swój Analizator Równań Różniczkowych w skrócie ARR. Pracowałem wtedy nad projektem EMAL-a, była to maszyna szeregową, zbudowana na logice opartej na starej technologii lampowej i opracowanej wcześniej przeze mnie wraz z Henrykiem Furmanem pamięci rtęciowej o pojemności 512 słów 40 bitowych umieszczonych w 32 rurach z rtęcią pracującej na częstotliwości 750 kHz. Maszyna pracowała w systemie dwójkowym w tempie 1500 operacji na sekundę. Niestety pracowała źle.

— **Dlaczego?**

— EMAL zbudowany został ze starych, a głównie poniemieckich, lamp elektronowych. Elementy te często się psuły. Dziś nazwalibyśmy to „barierą technologiczną” i być może zrezygnowali z dalszych prac. W tamtych czasach postanowiliśmy poszukać innych rozwiązań. EMAL-2 był już znacznie doskonalszy pod względem technicznym.

— **Czy EMAL-2 budował pan samodzielnie?**

— Nie — projektowałem go i konstruowałem wraz z Kazimierzem Balakerem, Lesławem Niemczyckim i Andrzejem Harlandem oraz technikami Henrykiem Furmanem, Gustawem Śliwickim, Stefanem Kostrzewą i Zbigniewem Grzywaczem.

EMAL-2 miał pamięć bębnową o pojemności 1024 słów rozmieszczonych na 32 ścieżkach. Ścieżka zerowa była pewnego rodzaju pamięcią ROM i zawierała prosty program wprowadzający. Cała logika i rejestry zbudowane były na magnetycznych elementach o bardzo wysokiej jak na owe czasy niezawodności. EMAL-2 miał hierarchiczną strukturę sterowania bardzo podobną do później opracowanych przez Petriego sieci. Maszyna zbudowana została ze standardowych elementów pamiętająco-logicznych. Do ich budowy wykorzystaliśmy plastikowe klocki, które obecnie noszą nazwę Lego, a w latach pięćdziesiątych pod nazwą „Młody Architekt” produkowała je Chemiczna Spółdzielnia Pracy „Świt”. Wewnątrz każdego klocka montowaliśmy układ elektroniczny. Potem można je było dowolnie składać i oczywiście w przypadku awarii bardzo łatwo wymienić.

Wydaje mi się, że jako pierwsi na świecie zastosowaliśmy takie rozwiązanie.

Nieźnym przykładem niezawodności EMAL-a było liczenie tablic funkcji Lagrange'a, które trwało trzy miesiące non stop bez najmniejszej awarii. Maszyna ta stała się załącznikiem Centrum Obliczeniowego Polskiej Akademii Nauk, przekształconego następnie w Instytut Podstaw Informatyki PAN, w którym obecnie pracuję razem z Leonem Łukaszewiczem. I proszę mi wierzyć — nie czuję się emerytem.

— **W przeszłości używano określeń „maszyny matematyczne”, „automaty liczące” zaś nauka, która zajmowała się nimi, nie posiadała nazwy. Profesor Władysław M. Turski w wydanej w roku 1980 książce pt. „Nie samą informatyką” napisał... „Z całą odpowiedzialnością mogę stwierdzić, że słowo „informatyka” zabrzmiało w Polsce po raz pierwszy ex cathedra w październiku 1968 roku. W Zakopanem na ogólnopolskiej konferencji poświęconej „maszynom matematycznym”, docent Romuald Marczyński wygłosił referat w którym... na dobre i złe zaproponował przyjęcie nazwy „Informatyka” wzorem istniejących już terminów *informatique* i *Informatik*”.**

— Nazwa była nam wtedy bardzo potrzebna. Do tej dziedziny przyznawali się matematycy, cybernetycy, elektrycy. Nazwanie dyscypliny naukowej, którą się zajmowaliśmy, umożliwiło nam samookreślenie niezbędne dla prawidłowego rozwoju.

Gdyby zapytał mnie pan, które ze swych dokonań uważam za najważniejsze, być może odpowiedziałbym — wprowadzenie do języka polskiego pojęcia „informatyka”. Zgodzi się pan chyba, że jak długo ludzie będą posługiwali się tym językiem słowo to, w takiej czy innej formie będzie żyło.

— **Słowo z całą pewnością — a nauka?**

— Mam nadzieję, że podobnie. Jednak jeżeli chcemy mieć postęp to musimy prowadzić prace badawcze.

Kiedy ja zaczynałem swoją karierę bardzo zależało mi na tym aby zrobić coś czego jeszcze nie było i tak np. w konstrukcji EMAL-a po raz pierwszy w świecie zastosowano specjalne mechanizmy dla przyspieszenia pobierania rozkazów. Nie było to wiele, ale moim zdaniem nauka, a w szczególności informatyka, rozwija się dzięki drobnym usprawnieniom.

Obecnie często słyszę pytanie stawiane naukowcom, czy ich prace badawcze przyniosą konkretny zysk, najlepiej w walucie wymiennej. Proszę pana, czy teoria względności ma swoją cenę, albo ile kosztuje twierdzenie Pitagorasa? W nauce inwestycje nie muszą przynosić szybkich zysków. Jeśli prowadzimy własne badania to możemy wymieniać się z innymi wynikami. Dzięki temu nie tracimy dysztansu do światowej czołówki.

— **Mam wrażenie, że już go straciliśmy.**

— Jeśli chodzi o budowę komputerów to z pewnością tak się stało. Między innymi z przyczyn o których mówiłem. Nasze opracowania teoretyczne nie są najgorsze co jednak nie rekompensuje naszego zacofania technologicznego. Na świecie liczą się dziś Amerykanie, Brytyjczycy, Francuzi, trochę Niemcy.

— **A Japończycy?**

— Nie! Oni jak na razie nie mają nic ciekawego do powiedzenia. Potrafią wspaniale zorganizować produkcję, ich towary są najwyższej jakości, ale głównie korzystają z gotowych rozwiązań opracowanych za granicą.

Być może w przyszłości zaskoczą nas też genialnymi pomysłami.

— **Jest pan człowiekiem, który od czterdziestu lat zajmuje się maszynami liczącymi. Jaki jest pana stosunek do nich?**

Pierwsze maszyny były oczywiście niedoskonałe, nie miały systemów operacyjnych, a konstruktorom zależało na tym aby w ogóle pracowały. Postęp technologii spowodował ogromny skok w miniaturyzacji komputerów, pojawiły się ogromne pamięci, doskonałe mikroprocesory. Jednak komputer to superidiota, bez człowieka niczego zrobi.

Uważam oczywiście, że jest to największy wynalazek od czasów wprowadzenia druku. Proszę nie pytać mnie jakie są perspektywy informatyki. Nikt chyba nie jest w stanie udzielić precyzyjnej odpowiedzi. W roku 1948 brytyjski matematyk Hartree przepowiadał, że jedna maszyna EDSAC (512 słów pamięci, 1000 operacji na sekundę, brak pamięci pomocniczej) wystarczy do zaspokojenia wszystkich potrzeb obliczeniowych Wielkiej Brytanii.

Nie chciałbym aby moją odpowiedź cytowano po latach w podobnym kontekście.

— **Dziękuję panu za rozmowę.**

Marek Czarkowski





## NOWA SERIA MOTOROLI

Znana amerykańska firma Motorola, producent 16-bitowych procesorów powszechnie stosowanych w takich komputerach jak Macintosh, Atari ST, Sinclair QL, itd., rozpoczęła produkcję nowej serii 32-bitowych układów opartych na architekturze typu RISC. Podstawowe chipy tej serii to MC88100 — jednostka centralna (CPU) i MC88200 — układ zarządzania pamięcią (CMMU — Cache/Memory Management Unit). Układ MC88100 zawiera 32 rejestry 32-bitowe, wbudowany arytmometr stało- i zmiennoprzecinkowy, rozszerzalną listę instrukcji i możliwość do rozbudowy architektury. Przy częstotliwości zegara 20MHz pojedynczy procesor osiąga wydajność 15 MIPS (ang. milion instructions per second). Maksymalna konfiguracja może zawierać cztery procesory pracujące współbieżnie. Pomocniczy układ MC88200 zawiera 16 kB szybkiej pamięci RAM i ułatwia zarządzanie pozostałą pamięcią komputera. Minimalny zestaw roboczy składa się z jednego układu MC88100 i dwóch układów MC88200. Celem zwiększenia wydajności systemu rozdzielono magistrale danych i adresów dla kodu programu i jego danych. Możliwe jest podłączenie ośmiu układów CMMU do jednego CPU, daje to 128kB szybkiej pamięci notatnikowej (ang. cache memory) dostępnej dla procesora. Zestawy te oferowane są przez firmę w postaci gotowych modułów o zastrzeżonej nazwie HYPERmodule. Ułatwiają one znacznie projektowanie nowych urządzeń, ponieważ rozwiązano w nich wiele problemów technicznych, z którymi stykają się projektanci, tworząc własne niezgodne wzajemnie standardy wy-

miany informacji między jednostką centralną a pamięcią. Interesująca jest także polityka software'owa Motoroli, ponieważ już na tak wczesnym etapie marketingowym zapewnia ona oprogramowanie rozwojowe i aplikacyjne dla opisywanej serii. Nie jest to typowy zwłaszcza w naszym kraju, gdzie wiele firm ogranicza się do sprzedaży samego sprzętu, pozostawiając użytkownikowi starania o programy, które pozwolą mu właściwie go wykorzystać. Obecnie dostępne są translatory większości języków programowania: C, FORTRAN, Pascal, Lisp, Prolog, COBOL i ADA. Dwie firmy amerykańskie Phoenix Technologies i Insignia Solution dostarczają oprogramowanie pozwalające na pracę programów aplikacyjnych pod kontrolą systemu operacyjnego MS-DOS na procesorze MC88100. Proponowane zastosowania serii 88000 dotyczą budowy systemów współbieżnych o dużej mocy przetwarzania, ze szczególnym uwzględnieniem komputerów mogących służyć do pracy nad zagadnieniami sztucznej inteligencji (AI) i projektowania wspomaganego komputerowo (CAD). Nowa seria Motoroli pojawiła się mniej więcej rok po wprowadzeniu Transputerów firmy INMOS, ale i obecnie układy te nie są jeszcze powszechne na rynku. Jednak w miarę upływu czasu stanowią one będą znaczącą konkurencję dla produktów INMOS'a, ponieważ Motorola dysponuje większym potencjałem finansowym i badawczym w stosunku do swego brytyjskiego konkurenta.

Oprac. BYTE 6,88  
(JM)

## PIRACI, STRZEŻCIE SIĘ!

**Na** 9 miesięcy więzienia, dodatkowo 2,5 roku w zawieszeniu oraz grzywnę w wysokości 10 tysięcy dolarów skazany został 14 lutego 1988 roku 18-letni mieszkaniec Chicago Herbert Zinn. Uznano go winnym nielegalnego włamania się do systemu komputerowego towarzystwa „American Telefon and Telegraph” powiązanego z Ministerstwem Obrony USA i krajami bloku NATO. Zinn piracko skopiował niektóre programy, nie było jednak wśród nich takich, które stanowiłyby tajemnicę państwową.

Wyrok ten jest jednym z pierwszych przypadków praktycznego zastosowania „Ustawy o oszustwach i nadużyciu techniki komputerowej”. Stwierdza ona, że za przestępstwo uważa się zamierzone, nieuprawnione włamanie się do programu komputera będącego własnością państwa, a także uszkodzenie go, w wyniku którego maszyna nie może we właściwy sposób wypełniać swoich funkcji. Za przestępstwo uważa się także zmianę, zniszczenie lub ujawnienie informacji, otrzymanej w wyniku nielegalnego dostępu do komputerowego banku danych. Za dokonanie tych przestępstw sąd może wymierzyć kary do 10 lat więzienia.

(ws)

# SZYBKI,

**Czy zastanawialiście się kiedyś nad szybkością swoich programów? Jeżeli tak, to na pewno przydadzą się Wam informacje, które znajdziecie w tym artykule. Jeżeli nie — przeczytanie go może Wam się przydać w przyszłości.**

Prędzej czy później każdemu zdarza się napisać program, którego wykonanie zajmuje bardzo dużo czasu. Oczywiście „bardzo dużo czasu” to pojęcie względne — raz będzie to kilkanaście minut, a raz kilka dni. Niezależnie od tego zależy nam na tym, żeby czas ten skrócić, i temu właśnie problemowi będzie poświęcony niniejszy artykuł. Nie będziemy się jednak zajmować algorytmami, w których leżą na ogół największe rezerwy — zajmujemy się gotowym kodem programu.

Na samym początku naszych rozważań musimy sobie uświadomić pewną rzecz. Wykonywanie programu przez komputer polega na kolejnym realizowaniu małych zadań, z których program się składa. Czas, jaki jest potrzebny na wykonanie programu jest równy sumie czasów potrzebnych do wykonania wszystkich tych drobnych zadań. Stąd wniosek, że program szybki to taki, w którym ilość operacji jest niewielka, i są to operacje możliwie mało czasochłonne. Czy już się domyślacie, jak trzeba poprawiać programy? W celu omówienia szczegółów tego działania posłużymy się przykładowym programem rysującym wykres pseudotrójwymiarowy (patrz PROGRAM1, który wraz z opisem był opublikowany w numerze Bajtka z sierpnia ubiegłego roku):

```
10 DIM M(255)
20 FOR Y=1 TO 141 STEP 5
30 FOR X=1 TO 141
40 LET Z=80*EXP(-1E-03*((Y-70)*(Y-70)+(X-70)*(X-70)))
50 LET X1=X+Y*COS(PI/4)
60 LET Y1=Z+Y*SIN(PI/4)
70 IF Y1>=M(X1) THEN LET M(X1)=Y1: PLOT X1,Y1
80 NEXT X
90 NEXT Y
```

Nie zajmując się zastosowanym algorytmem, zajmiemy się wyłącznie stopniowym przyspieszaniem programu, przy pomocy wymienionych poniżej reguł. Zaczniemy od

### REGUŁY PIERWSZEJ,

mówiącej, że optymalizację programu należy zacząć od najbardziej wewnętrznych pętli. Jest to oczywiste — operacje, które się tam znajdują, są wykonywane najczęściej, toteż każdy, nawet najmniejszy zysk czasu, jaki uda się osiągnąć będzie wielokrotniony. W naszym przypadku optymalizację trzeba zacząć od linii 40—70, które są wykonywane w czasie działania programu 4089 razy.

### REGUŁA DRUGA

uściśla, od czego powinniśmy zacząć pracę. Jeżeli w programie występują kilkakrotnie identyczne wyrażenia, to warto je zastąpić zmienną, której wartość będzie obliczana tylko jeden raz. Należy tak postąpić również ze wszystkimi stałymi. Popatrzmy teraz na nasz program. Na pierwszy rzut oka widać, że zupełnie niepotrzebnie za każdym razem liczymy w liniach 50 i 60 wartość  $\sin(\pi/4)$ . Zaczniemy więc od dopisania do programu linii:

```
15 LET A=COS(PI/4)
i zamienienia linii 50 i 60 na:
50 LET X1=X+Y*A
60 LET Y1=Z+Y*A
```

(Skorzystaliśmy tu z zależności  $\sin(\pi/4)=\cos(\pi/4)$ , dzięki czemu wystarczyło wprowadzenie jednej dodatkowej zmiennej. Dla innej wartości kąta musielibyśmy wprowadzić jeszcze jedną zmienną, służącą do przechowywania wartości sinusa). Przy pomocy tej prostej poprawki zaoszczędziliśmy 8177 dzieleni i tyle samo obliczeń wartości funkcji trygonometrycznej. Jest to jednak dopiero początek drogi. Po co w sąsiadujących ze sobą liniach 50 i 60 dwukrotnie wykonywać to samo mnożenie? Wprowadźmy nową zmienną E i dopiszmy do programu linię:

```
45 LET E=A*Y,
oczywiście zmieniając przy okazji linie 50 i 60 na:
50 LET X1=X+E
60 LET Y1=Z+E
```



## SZYBSZY, NAJSZYBSZY

Tym sposobem zaoszczędziliśmy 4089 mnożeń. Ale czy na pewno zrobiliśmy to dobrze? Reguła pierwsza mówi o optymalizacji najgłębiej zagnieźdzonych pętli, i można ją rozumieć również jako konieczność przeniesienia na zewnątrz pętli tych obliczeń, które wystarczy wykonać jeden raz. Wartość naszej nowej zmiennej E zależy tylko od wartości zmiennej Y, a to znaczy, że wartość E wystarczy policzyć raz w pętli 20—90, czyli 29 razy w ciągu wykonywania całego programu. U nas będzie ona liczona 4089 razy. Jaka jest na to rada? Bardzo prosta — to co jest linią 45 powinno się znaleźć poza wewnętrzną pętlą z linii 30—80, na przykład jako linia:

```
25 LET E=A*Y
```

co pozwala zlikwidować 4060 mnożeń.

Co możemy jeszcze zrobić? Sprawdźmy, czy reguły drugiej nie da się zastosować do linii 40-tej. Z tych samych powodów, dla których przeniesiliśmy mnożenie  $A*Y$  z linii 50 i 60 do 25-tej, możemy zastąpić iloczyn  $(Y-70)*(Y-70)$  nową zmienną C, której wartość policzymy „na raty”, zaoszczędzając za każdym razem jedno odejmowanie:

```
26 LET C=Y-70
```

```
27 LET C=C*C
```

(Zauważcie, że znowu przeniesiliśmy obliczenia z pętli wewnętrznej do pętli zewnętrznej — jest to bardzo często stosowany chwyt.) Linia 40 wygląda teraz tak:

```
40 LET Z=80*EXP(-1E-03*(C+(X-70)*(X-70)))
```

i aż się prosi o wykonanie analogicznej operacji ze zmienną X:

```
34 LET D=X-70
```

```
40 (LET Z=80*EXP(-1E-03*(C+D*D)))
```

co pozwala na zaoszczędzenie następnych 4089 odejmowań.

Nasz program wygląda w tej chwili następująco:

```
10 DIM H(255)
```

```
15 LET A=COS(PI/4)
```

```
20 FOR Y=1 TO 141 STEP 5
```

```
25 LET E=A*Y
```

```
26 LET C=Y-70
```

```
27 LET C=C*C
```

```
30 FOR X=1 TO 141
```

```
34 LET D=X-70
```

```
40 LET Z=80*EXP(-1E-03*(C+D*D))
```

```
50 LET X1=X+E
```

```
60 LET Y1=Z+E
```

```
70 IF Y1>=H(X1) THEN LET H(X1)=Y1: PLOT X1,Y1
```

```
80 NEXT X
```

```
90 NEXT Y
```

a oszczędności wynoszą 8177 dzieleni, tyle samo obliczeń sinusa i cosinusa, 12209 mnożeń i 12238 odejmowań. (Radzę dokładnie prześledzić, skąd się wzięły te liczby). Na Spectrum oznacza to skrócenie czasu potrzebnego na wykonanie programu z 13 minut 26 sekund do 6 minut 55 sekund. Jest to właściwie koniec przyspieszania naszego programu, co nie oznacza końca artykułu. Będą jeszcze trzy reguły, najpierw

### REGUŁA TRZECIA

doradzająca nam takie przekształcenie wszystkich wyrażeń, aby ich obliczenie wymagało jak najmniejszej ilości operacji. Do tej pory rozbijaliśmy różne wyrażenia na mniejsze kawałki, by zmniejszyć ilość obliczeń, co nie zawsze jest możliwe. Na przykład przy liczeniu wartości wielomianu jedynym sposobem na zminimalizowanie ilości obliczeń jest posłużenie się odpowiednim przekształceniem, czyli zastosowanie schematu Hornera, o którym była już w Bajtku mowa (4/88). Dla przypomnienia — obliczenie wartości wielomianu  $ax^3+bx^2+cx+d$  najszybciej wykonuje się jako  $x(x(ax+b)+c)+d$ , i nie ma możliwości zmniejszenia ilości potrzebnych mnożeń i dodawań poniżej liczby wynikającej z zastosowanego przekształcenia — dla wielomianu trzeciego stopnia konieczne są trzy mnożenia i tyleż dodawań.

Przedostatnia

### REGUŁA CZWARTA

dotyczy zastępowania niektórych operacji innymi, dającymi ten sam efekt w znacznie krótszym czasie. Wiadomo że dodawanie jest szybsze od

mnożenia, a mnożenie od dzielenia i potęgowania, toteż zamiast pisać w programie:

```
LET A=B/2+C↑2+2*D
```

dużo lepiej jest napisać:

```
LET A=0.5*B+C*C+D+D
```

Oczywiście nie należy przesadzać z takimi chwytami —  $C↑6$  zastąpione przez iloczyn  $C*C*C*C*C*C$  będzie na ogół wolniejsze, ale to już jest kwestia stosowanego kompilatora (lub interpretera).

### REGUŁA PIĄTA

dotyczy instrukcji warunkowych, a dokładniej rzecz biorąc sprawdzania, czy daną instrukcję wykonać czy nie. Przez odpowiednie zorganizowanie programu możemy tu też trochę zyskać. Posłużmy się prostym przykładem — zamieńmy znak ASCII, zapamiętany w zmiennej A\$, na wielką literę, jeśli jest literą małą, w przeciwnym przypadku nie ruszajmy go:

```
10 IF A$="a" AND A$<="z" THEN LET A$=CHR$(CODE(A$)-32)
```

W tak napisanym programie warunek  $A$<="z"$  będzie sprawdzany zawsze. Tymczasem jeśli pierwszy warunek —  $A$="a"$  — był fałszywy wiadomo, że całe wyrażenie  $A$="a" AND A$<="z"$  musi mieć wartość „fałsz”, i sprawdzanie drugiego warunku jest niepotrzebne. Dlatego też można postąpić tak:

```
10 IF A$="a" THEN IF A$<="z" THEN  
LET A$=CHR$(CODE(A$)-32)
```

lub tak:

```
10 IF A$="a" THEN GO TO 30
```

```
20 IF A$="z" THEN LET A$=CHR$(CODE(A$)-32)
```

```
30 ...
```

Za każdym razem możemy zaoszczędzić (w zależności od tego czy pierwszy warunek jest spełniony czy nie) czas potrzebny na sprawdzenie drugiego warunku. Z różnych względów pierwszy sposób jest lepszy i czasem bywa szybszy.

Ostatnia praktyczna rada, nie wyodrębniona jako kolejna reguła, dotyczy używania pętli — zwłaszcza krótkich, takich w których wykonuje się niewielką liczbę operacji. Otwarcie pętli i wykonywanie operacji związanych z jej zamknięciem, to znaczy zmiana wartości zmiennej sterującej i sprawdzanie, czy wykonywanie pętli nie powinno zostać zakończone, wymagają trochę czasu, na który nie mamy żadnego wpływu. Dlatego też czasem warto pewne operacje rozisać — na przykład zamiast

```
10 LET A=0  
20 FOR I=1 TO 7  
30 LET A=A+B(I)  
40 NEXT I
```

warto napisać

```
10 LET A=B(1)+B(2)+B(3)+B(4)+B(5)+B(6)+B(7)
```

Dotyczy to również różnych pętli, w których są drukowane wyniki lub rysowane ramki albo inne wzorki — jeśli zależy nam na czasie, warto zrezygnować ze zwartości programu.

Przytoczone reguły są bardzo ogólne, i w zasadzie ich zastosowanie powinno pozwolić na przyspieszenie każdego programu, niezależnie od języka w jakim został on napisany. Niestety, czasem zdarza się, że zmniejszając ilość obliczeń, zwiększamy nieświadomie ilość operacji, które będą wykonane przez komputer. Dotyczy to zwłaszcza języków interpretowanych, takich jak BASIC. Interpretery często zachowują się jak pedantyczny człowiek, który trafiając na coś, co trzeba policzyć, sięga do kieszeni, wyciąga kalkulator, włącza go, wykonuje obliczenia, wyłącza kalkulator i chowa go z powrotem. Czas potrzebny na wykonanie wszystkich operacji nie związanych bezpośrednio z samym liczeniem może być tak długi, że rozbicie jednej instrukcji:

```
LET A=(X-70)*(X-70)
```

na dwie:

```
LET A=X-70
```

```
LET A=A*A
```

zamiast zysku przyniesie straty — jedno odejmowanie jest krótsze, niż wszystkie czynności związane z uruchamianiem kalkulatora. Żeby nie być gołosłownym, przytoczę dwie liczby — czas potrzebny Spectrum na tysiąckrotne wykonanie pierwszej wersji obliczeń wynosi 12.19 sekundy, dla drugiej — 12.70 sekundy. Podobnie zachowują się i inne interpretery, np. BASICA na IBM PC — druga wersja obliczeń jest wprawdzie szybsza, ale tylko o 2%. Tutaj może już pomóc tylko dobra znajomość używanego języka, i proste próby, podobne do opisanych przed chwilą (podczas ich przeprowadzania dobrze jest pamiętać o tym, że wykonywanie tego samego działania na różnych danych zajmuje różną ilość czasu).

Oprócz omówionych istnieją również oczywiście i inne metody przyspieszania działania programów — niektóre bardziej ogólne, inne związane z konkretnymi językami lub nawet konkretnymi kompilatorami (patrz np. artykuł A. Kadlofa „W pogoni za mikrosekundami”, Komputer 8/88). Nie sposób w krótkim artykule omówić wyczerpująco tak szerokiego tematu. Mam jednak nadzieję, że udało mi się pokazać kilka podstawowych chwytów, które warto jest znać, nawet jeśli pisuje się programy wyłącznie na własny użytek.

Na koniec pewna uwaga dotycząca stylistyki programów. Programy działające szybko z reguły są nieczytelne — kto ma wątpliwości niech porówna nasz przykład w wersji początkowej i po dokonaniu w nim poprawek. Tym ważniejsze staje się w takim wypadku umieszczenie w programie odpowiednich komentarzy, pozwalających na zrozumienie skąd w linii 40 wzięły się zmienne C i D i co one oznaczają.

Marcin Borkowski



# EDYTORY, EDYTORY...

Komputery Atari to nie tylko drogie zabawki, jak sądzi duża część użytkowników. Bogaty wybór oprogramowania użytkowego pozwala zastosować je także w pracy. Czasami trudno jest zdecydować się, który program wybrać. Postaramy się pomóc w tym wyborze naszym Czytelnikom.

Najczęstszym zastosowaniem wszystkich komputerów jest redagowanie tekstów. Stąd dla każdego komputera opracowano po kilkanaście różnych edytorów. Opiszę tu dziewięć najpopularniejszych edytorów tekstu dla Atari XL/XE. Oczywiście ocena ich wartości jest moim subiektywnym odczuciem i nie każdy musi się z nią zgadzać.

## AtariWriter+

**AtariWriter+** (w tabeli AW+) jest poprawioną wersją jednego z pierwszych edytorów Atari i nadal należy do najlepszych programów tego typu. Jego zaletą jest współpraca z dowolnymi drukarkami oraz możliwość przeglądania strony wydruku o szerokości do 200 kolumn. Przy wyszukiwaniu i wymianie fragmentów tekstu można użyć znaku „?” jako jokera, który zastępuje każdy inny znak. Ponadto podkreślone części tekstu są zaznaczane przez wyświetlenie w negatywie, nie ma jednak wyróżnienia druku wytłuszczonego i kursywy. Największą wadą jest mały rozmiar bufora tekstu na komputerach 800XL i 65XE,

dla 130 XE jest on natomiast największy. Pewnym ograniczeniem jest także maksymalna liczba 255 możliwych do dołączenia rekordów z listy SynFile+. Producentem programu jest Atari Corp.

## PaperClip

**PaperClip** (PC) jest na Zachodzie uznawany za najlepszy edytor dla ośmiobitowych Atari (posiada on także wersję na Atari ST). Ma bardzo duże możliwości redakcyjne, a polecenia są szybkie, efektywne i łatwe do zapamiętania. Do najważniejszych zalet należą: jednoczesne redagowanie dwóch dokumentów, funkcje Macro i Undo, wbudowane funkcje matematyczne, automatyczne zapisywanie redagowanego dokumentu, współpraca ze wszystkimi drukarkami, możliwość włączania grafiki w tekst i użycia dowolnego kroju liter (jeśli można je zdefiniować w drukarce) oraz duże i wyraźne litery na ekranie (tryb ANTIC 3). Długość wiersza na ekranie jest definiowana przez użytkownika i może być ustawiona zgodnie z późniejszym wydrukiem. Pewną wadą jest stosunkowo długi czas trwania operacji wymiany fragmentów tekstu w dużych dokumentach, lecz za to można jednocześnie wymieniać sześć różnych ciągów. Ponadto na dyskietce znajduje się specjalny program do automatycznej konwersji plików zapisanych przez **AtariWriter+** na format **PaperClipa**. Producentem tego programu jest Batteries Included.

## SpeedScript

**SpeedScript** (SS) jest stosunkowo krótkim i prostym edytorem autorstwa

Charlesa Branona, posiadającym jednak spore możliwości redakcyjne. Zamiast przeglądania tekstu posiada możliwość wydruku na ekran (S:), lecz tylko w 40 kolumnach. Nie ma wbudowanych kodów drukarek i uzyskanie dodatkowych efektów wymaga wpisywania kodów bezpośrednio w tekst. Największymi jego zaletami są duży bufor i możliwość pracy z magnetofonem. Ponadto tekst na ekranie jest wyraźny, dzięki użyciu trybu ANTIC 3.

## Word Magic

**Word Magic** (WM) jest programem napisanym przez Blue Collar Software dla amerykańskiego miesięcznika „Antic”. Może pracować z dowolną drukarką i umożliwia wydruk grafiki (ponadto zawsze występuje w połączeniu z programem graficznym **Graphic Magic**). Przegląd tekstu odbywa się w trybie graficznym (znak = punkt), więc można sprawdzić tylko ogólny układ tekstu na stronie. Do druku i przeglądania odczytuje z dyskietki specjalny program, a dokument również musi być odczytywany z dyskietki. Przesuwanie kursora przy pomocy klawiatury i/lub joysticka.

## HomeText

**HomeText** (HT) stanowi część pakietu **HomePak**, którego pozostałymi elementami są baza danych **Home-**

**Find** i program komunikacyjny **HomeTerm**. Mała pojemność bufora pozwala na pisanie dokumentów nie przekraczających 8 stron znormalizowanego maszynopisu. Jest bardzo łatwy w użyciu, natomiast denerwujący i mało czytelny jest stylizowany krój liter wyświetlanych na ekranie. Producentem programu jest firma Batteries Included.

## HomeWord

**HomeWord** (HW) to program firmy Sierra On-Line przystosowany przede wszystkim dla drukarek Atari. Bardzo mały bufor umożliwia zredagowanie czterech stron maszynopisu. Sterowany jest z obrazkowego menu i posiada wbudowaną funkcję HELP, dzięki czemu nie wymaga praktycznie nauki. Przy zapisie automatycznie wykonuje kopię pliku. Przegląd strony jest prowadzony na bieżąco w trybie graficznym, co predestynuje go do przygotowywania różnego rodzaju tabel. Niestety posiada własny format zapisu na dyskietce i jego plik nie może być odczytany przez inne edytory.

## First Xlent

**First XLEnt** (FX) jest najnowszym edytorem Atari. Posiadanyymi możliwościami redakcyjnymi ustępuje tylko **PaperClipowi**. Z wymienionych tam możliwości nie posiada automatycz-

EDYTOR	AW+	PC	SS	WM	HT	HW	FX	ST	PT
nośnik	D	D	D/K	D	D	D	D	D	D/K
tekst w XL (KB)	12.3	25.0	27.9	23.7	6.5	3.9	28.3	19.5	12
text w XE <sup>1</sup>	46.5	30	-	RD	-	-	RD	RD	-
sterowanie z menu	T	N	N	T	T	T	T	N	T
funkcja HELP	N	T	N	T	N	T	T	N	N
zapis automat.	N	T	N	N	N	N	N	N	N
druk części pliku	T	T	T <sup>2</sup>	N	N	N	T	N	N
funkcja undo	N	T	N	N	N	N	N	N	N
liczba okien	1	2	1	1	1	1	2	1	1
przesuw poziomy	T	T	N	N	N	N	N	T	T
odstęp proporc.	N	T	N	N	N	N	T	N	T
nagłówek/stopka	T	T	T	T	T	T	T	T	N
wierszy/nagł., st.	2	3	2	1	2	1	1	1	-
druk 2-kolumnowy	T	T	N	N	N	N	T	N	N
wbud. kody druk.	T	T	N	T	N	N	N	T	N
listy z SynFile+	T	T	N	T	T	N	T	N	N
odczyt innych	T	T	T	T	T	N	T	T	T
przegląd wydruku <sup>3</sup>	40	40	N	GR	N	GR	80	80	N
druk grafiki	N	T	N	T	N	N	T	T	N
polskie litery	N <sup>4</sup>	T	N <sup>4</sup>	N	N	N	T	T	T

UWAGI:

1. Minus = program nie rozpoznaje wersji, RD = Ramdisk.
2. Tylko od początku dowolnej strony do końca pliku.
3. GR = tryb graficzny, 40 i 80 = liczba kolumn na ekranie.
4. Istnieje wersja z polskimi literami dla Atari 1029.



Z kraju otrzymuję regularnie „Bajtka”, „IKS”, „Komputer” oraz „Mikroklan”. Jestem więc zorientowany, co interesuje użytkowników

**małego Atari. Stwierdziliście, że nie posiadacie bliższych danych o stacji XF551, ani jej zdjęcia. Przesyłam Wam wycinek reklamujący tę stację w „Atari User” oraz ofertę rozszerzeń do 1050.**

**Artur Sikora  
Katerini, Grecja**

Publikujemy nadesłane zdjęcie, a pozostałe informacje podajemy poniżej.

Parametry Atari XF551:  
— zapis dwustronny w podwójnej gęstości;  
— łatwe dołączenie i prosta obsługa;  
— współpraca z komputerami 800XL, 65XE i 130XE;  
— obudowa dostosowana wzorniczo do serii XE.

Ponadto oferta zawiera:  
— 1050 Super Archiver II: kopiuje wszystkie programy (bez wyjątku), formatuje do 40

(!) sektorów na ścieżce, drukuje zawartość ekranu i wiele innych możliwości (60 funtów);

— Silencer: zmniejsza o ponad 50% hałas wytwarzany przez stację i pozwala na zapisywanie zabezpieczonych dyskietek (18 funtów);

— Richman's Word Processor: 80 kolumn tekstu na ekranie i duże możliwości redakcyjne (25 funtów);

— Controller Card II: pozwala na korzystanie z programów użytkowych, rozszerzeń i bibliotek podczas pracy do-

# ATARI XF 551



nego zapisu, funkcji matematycznych oraz funkcji Macro i Undo. Umożliwia natomiast redagowanie tekstu w buforze kopiowania (maksymalnie 800 znaków). Poszukiwanie i wymiana fragmentów tekstu odbywają się niemal natychmiast. Za wadę należy jednak uznać powrót z menu na początek tekstu i w tryb zastępowania (replace). Jak wszystkie programy XLEnt Software może współpracować z dowolnymi drukarkami i przy użyciu dowolnych krojów liter — jako jedyny edytor Atari wyświetla na ekranie polskie litery (pomijam tu przeróbki **AtariWriter** i **SpeedScript** przeznaczone wyłącznie dla Atari 1029).

## StarTexter

**StarTexter** (ST) jest programem niemieckiego wydawnictwa SYBEX-Verlag i posiada wersję polską wykonaną w kraju. Umożliwia użycie polskich liter jako definiowalnych znaków graficznych (zob. „Komputer” 9/88). Pozwala na współpracę z większością drukarek. Dostępnymi możliwościami redagowania tekstu ustępuje jednak programom amerykańskim (**PaperClip**, **AtariWriter+** i **First XLEnt**). Ponadto ma stosunkowo nieduży bufor — niecałe 20 KB. Zaletą jest natomiast praca na pełnej 80 kolumnowej stronie, choć widoczna jest tylko jej część.

## PioText

**PioText** (PT) jest opracowanym w Polsce przez Piotra Szulkiewicza edytorem przeznaczonym przede wszystkim do współpracy z magnetofonem. Jest to jego podstawowa zaleta. Możliwości redakcyjne programu są bardzo małe — brak jest nawet tak podstawowych funkcji, jak wyszukiwanie fragmentów tekstu. Niewielki jest także rozmiar redagowanego dokumentu. Oczywiście — jak przystało na polski program — na ekranie i drukarce uzyskuje się polskie litery i to jest jego druga zaleta. Niestety innych nie udało mi się stwierdzić.

## Podsumowanie

Ważniejsze parametry opisanych edytorów zostały zestawione w zamieszczonej tabeli. Wybór należy już do użytkowników, zależy bowiem w dużym stopniu od potrzeb i indywidualnych upodobań. Ze swej strony polecałbym dla posiadaczy stacji dysków **PaperClip** i **First XLEnt** (tych właśnie sam używam), a dla posiadaczy magnetofonów **SpeedScript** i ewentualnie pierwszą wersję **AtariWriter** (bez plusa).

Wojciech Zientara

W znajdującym się w tym numerze artykule „Szybki, szybszy, najszybszy” Marcin Borowski opisuje ogólne zasady przyspieszania pracy programu. Każdy interpreter ma jednak swoją odrębną specyfikę i dlatego należy do każdego komputera podchodzić indywidualnie. Zastanówmy się więc, jak przyspieszyć działanie programu na Atari.

Zacznijmy od obliczeń. Zasadą jest, że dla zaoszczędzenia miejsca w pamięci często stosowane stałe zastępuje się zmiennymi. Niestety spowalnia to znacznie program. Gdy interpreter napotka wartość stałą, to przenosi ją do bufora operacji arytmetycznych i wykonuje działanie. W przypadku zmiennej musi dodatkowo według podanego numeru odszukać wartość zmiennej w tablicy. Konieczny jest więc rozsądny kompromis między długością i szybkością programu.

Największe rezerwy tkwią zwykle w strukturze programu. Przy wykonywaniu instrukcji GOTO, GOSUB i RESTORE interpreter znajduje wskazany numer wiersza przeszukując program od początku. Trzeba więc najczęściej używane procedury umieszczać na początku programu. Z instrukcjami DATA jest trochę inaczej, gdyż warto je przenosić na początek tylko wtedy, gdy jest ich niewiele i zawierają niewielką liczbę danych. Zapytanie, gdzie umieścić początek programu. Najlepiej na końcu, bowiem jest wykonywany tylko raz i to wtedy, gdy zwykle nie zależy nam na szybkości.

Ponieważ czas przeszukiwania programu zależy od liczby wierszy, a ich długość jest nieistotna, to czym więcej instrukcji zmieścimy w jednym wierszu programu, tym będzie on szybszy. Dodatkowe przyspieszenie uzyskamy usuwając z programu wszystkie komentarze, ale wtedy trzeba zachować jedną kopię z komentarzami na wypadek, gdyby zaszła konieczność dokonania w programie poprawek (czyli prawie zawsze).

Instrukcje GOTO i GOSUB stwarzają jeszcze inne problemy. W wykonaniu różnią się one jedynie zapamiętywaniem (lub nie) adresu powrotnego. Prosty wniosek, że GOTO jest szybsze. Ale dwie instrukcje GOTO są znacznie wolniejsze niż para GOSUB/RETURN. W ogóle instrukcje GOTO i GOSUB należą do najwolniejszych (poza obliczeniami). Nie wolno jednak unikać

GOSUB, gdyż pozwala ona za zasadnicze skrócenie całego programu.

Znacznie szybsza od GOSUB jest instrukcja FOR/NEXT. Powinno się więc zamienić je tam, gdzie to możliwe, lub przynajmniej przenieść pętlę do procedury i usunąć wywołania z pętli. Przy opuszczaniu pętli i procedur w sposób inny niż naturalny (GOTO zamiast NEXT lub RETURN) nie wolno zapominać o użyciu instrukcji POP. Kilka takich miejsc w programie może spowodować zajęcie całej pamięci komputera na stos bieżący Basica. Przede wszystkim każde następne otwarcie pętli lub wywołanie procedury będzie wolniejsze od poprzedniego. Z tego powodu należy również ograniczać liczbę kolejnych zagnieżdżeń pętli i procedur (oczywiście tak, aby nie ucierpiał na tym program).

Niewielkie oszczędności można także uzyskać zastępując instrukcje SETCOLOR przez POKE. Komputer ma za każdym razem do odczytania i „przetrawienia” o jedną liczbę mniej — same instrukcje są wykonywane ze zbliżoną szybkością, a wielkość liczby nie ma tu znaczenia. Podobnie stosowanie „IF... THEN numer” zamiast „IF... THEN GOTO numer” daje jedną instrukcję do wykonania mniej. W tym przypadku po napotkaniu liczby skok jest wykonany natychmiast, zaś instrukcja GOTO musi być dodatkowo rozpoznana przez interpreter.

Największe możliwości przyspieszenia programu w Atari krają się jednak poza interpreterem Basica. Otóż podczas pracy komputera do 30% czasu zabiera tworzenie obrazu. Jeśli obraz nie jest konieczny, np. przy obliczeniach, to należy go wyłączyć instrukcją POKE 559,0, a po zakończeniu obliczeń ponownie włączyć przez POKE 559,34. Trzeba jednak pamiętać, że nie zauważymy wtedy wystąpienia błędu i możemy długo czekać na wynik. Niezbędne jest w takim przypadku zastosowanie instrukcji TRAP, która będzie powodować włączenie obrazu i sygnalizowanie błędu.

Gdy rodzaj programu wymaga stałego wyświetlania obrazu, zmniejszenie zużywanego przezeń czasu uzyskamy stosując tryby graficzne o mniejszej rozdzielczości. W tym przypadku miernikiem jest długość programu ANTIC-a i liczba bajtów danych w pamięci obrazu. Najwięcej czasu zabierają tryby 8—11, 14 i 15, najmniej tryby 1—3, 12 i 13.

Oczywiście nie wyczerpaliśmy tu wszystkich możliwości — na przykład obliczenia funkcji trygonometrycznych w radianach są krótsze o kilkanaście rozkazów od obliczeń w stopniach. Wykorzystanie dalszych sposobów przyspieszania programu jest jednak nieco trudniejsze i dalej znacznie mniejsze efekty.

Wojciech Zientara

wolnego programu — może go zatrzymać w dowolnej chwili (60 funtów), ponadto na dyskietce dodawane jest kilka programów użytkowych, m.in. 14 zestawów znaków, menu w oknach, itd.;

— Turbobase: pakiet zintegrowany zawierający edytor tekstu, arkusz kalkulacyjny, bazę danych, kalendarz i program graficzny, możliwość uzyskania 80 kolumn, współpraca z rozszerzeniami pamięci i wszystkimi stacjami dysków (również z twardym dyskiem), podręcznik 700 stron (120 funtów).

## — TURBO — DLA MAGNETOFONU

Często w naszym piśmie ukazują się reklamy różnych firm, które oferują przeróbkę magnetofonu. Ma to umożliwić około dziesięciokrotne zwiększenie szybkości transmisji, nadawanie nazw plikom i inne ułatwienia. Wielu Czytelników

pyta nas, czy warto dokonać takiej przeróbki i którą z licznych ofert wybrać.

Na te pytania redakcja **NIE MOŻE** udzielić informacji. Żadna z firm nie zwróciła się do nas z prośbą o przetestowanie lub ocenę swojego rozwiązania i przedstawienie tej oceny na łamach pisma. Mogą być trzy przyczyny takiej sytuacji.

— Po pierwsze, obawa przed nadmierną reklamą, ale przeczą temu reklamy w dziale „Wszystko dla wszystkich”.

— Po drugie, brak „mocy przerobowych” i obawa, że nie da się zrealizować wszystkich zamówień.

Po co więc ogłoszenia w „Bajtku”.

— Po trzecie, może po prostu przeróbka magnetofonu nie jest tak wspaniała, jak sugerują to reklamy i wykonawcy boją się weryfikacji przez naszą redakcję.

Trzeba na koniec wyjaśnić, dlaczego sami nie zajmujemy się tą sprawą, nie czekając na producentów. Otóż wszyscy uważamy, że komputer z magnetofonem (nawet najlepszym) jest tylko zabawką i jedynie fanatycy mogą używać go do pracy. Najlepszą przeróbką magnetofonu jest zamienienie go na stację dysków i to właśnie polecamy rozważyć Czytelników.

Marek Zachar





Po zapoznaniu się z budową mikroprocesora 6502 powinniśmy rozpocząć naukę programowania. Niestety jest to jeszcze niemożliwe — potrzebne jest do tego celu narzędzie, czyli program tłumaczący mnemoniki (nazwy rozkazów) na kody języka maszynowego. Programy takie dzielą się na dwie podstawowe grupy: asemblery i monitory. Musimy się najpierw zająć nimi, a wszystkich niecierpliwych odsyłam do „Klanu Commodore” (procesory w Atari i Commodore są prawie identyczne).

Asemblery są to programy tłumaczące program źródłowy zapisany w formie czytelnej dla człowieka na ciąg liczb kodu maszynowego, zwany programem wynikowym. Wewnętrznie (bo zwykle użytkownik tego nie widzi) składają się one z edytora i kompilatora. Pierwszy umożliwia napisanie programu w assemblerze, czyli przy pomocy mnemoników. Drugi służy do zamiany tego programu na kod procesora. Ponadto muszą posiadać zespół procedur pozwalających na zapisywanie i odczytywanie programów, zarówno źródłowych, jak i wynikowych. Większość assemblerów umożliwia dodatkowo wykonanie bezpośrednio na pamięci komputera prostych operacji, np. odczytu i zapisu pojedynczych bajtów. Natomiast monitory, zwane także programami uruchamiającymi, są skonstruowane niejako odwrotnie: służą przede wszystkim do wykonywania operacji na pamięci komputera (łącznie z wykonywaniem i śledzeniem zawartego w niej programu), zaś możliwość pisania programu jest funkcją pomocniczą.

W przypadku assemblerów, tak jak w każdej grupie programów, istnieje duży wybór. Wszystkie korzystają z takich samych mnemoników, zaś ich działanie jest bardzo podobne. Z całego tego bogactwa wybrałem więc jeden assembler — najpopularniejszy w Polsce i stale przeze mnie wykorzystywany — **MAC/65** renomowanej firmy OSS. Jest on dostępny na wszystkich trzech nośnikach: dyskietce, kasecie i cartridge'u.

## WIADOMOŚCI OGÓLNE

**MAC/65** posiada dwa tryby pracy: składniowy i tekstowy. Mają one tylko jedną różnicę. W trybie składniowym po wprowadzeniu każdego wiersza programu sprawdzana jest jego poprawność, a ewentualne błędy są sygnalizowane tak jak w Atari Basic. W trybie tekstowym nie ma kontroli poprawności programu, można więc wykorzystać go do opracowywania programów napisanych przy pomocy innych assemblerów oraz do pisania programów w innych językach (takich, które wymagają numerowania wierszy programu). Rozróżnienie trybu pracy jest możliwe dzięki komunikatowi gotowości (jak **READY** w Basicu): w składniowym jest to **EDIT**, a w tekstowym **TEXTMODE**.

Po uruchomieniu **MAC/65** znajduje się zawsze w trybie składniowym. Przejście do trybu tekstowego wykonuje się poleceniem **TEXT**. Polecenie **NEW** powoduje powrót do trybu składniowego. Każda zmiana trybu pracy assemblera powoduje SKASOWANIE zawartego w pamięci programu, trzeba więc pamiętać o zapisaniu go przedtem.

Do zakończenia pracy z assemblerem służą trzy polecenia: **BYE** powoduje przejście do programu testującego, a **DOS** do DOS-u (jak w Basicu). Polecenie **CP** (Command Processor) ma działanie identyczne jak **DOS**. Przy pracy z DOS XL, dla którego **MAC/65** został zaprojektowany, po wyjściu do DOS-u pamięć nie jest kasowana. Wpisanie (w DOS XL) polecenia **RUN** lub gwiazdki (\*) ponownie uruchamia assembler bez zniszczenia opracowywanego programu źródłowego.

Format wprowadzanego wiersza programu jest jednokrotny, niezależnie od trybu pracy. Edytor **MAC/65** rozpoznaje małe i duże litery oraz znaki w negatywie. Zostają one jednak zawsze doprowadzone do standardowej postaci, z wyjątkiem znaków zawartych w cudzysłowach i w komentarzu. Każdy wiersz zawiera kolejno: numer, etykietę, mnemonik i komentarz. W poprawnym wierszu musi wystąpić numer i co najmniej jeden z pozostałych elementów. Wpisanie samego numeru powoduje usunięcie z pamięci wiersza o takim numerze.

Numer wiersza programu może przyjmować wartości od 0 do 65535 i musi być oddzielony od dalszej treści co najmniej jedną spacją. Bezspośrednio po tej spacji rozpoczynają się etykiety, które służą do oznaczania wierszy programu. Etykieta jest nazwą i może składać się z liter, cyfr oraz znaków „?”, „@” i kropki (.), przy czym cyfra i kropka nie może być pierwszym znakiem. Etykiety rozpoczynające się od znaku „?” są traktowane jako lokalne. Etykieta może zawierać do 127 znaków (wszystkie są znaczące). Mnemonik rozkazu musi być oddzielony co najmniej jedną spacją od etykiety lub, jeśli nie ma jej w wierszu, dwiema spacjami od numeru wiersza. Jeśli rozkaz posiada operand (wartość, na której wykonywana jest operacja), to spacja między rozkazem i operandem jest zbędna. Komentarz powinien rozpoczynać się średnikiem (;) i może być wpisywany bez odstępów, z wyjątkiem jednej spacji po numerze wiersza. Jeżeli komentarz rozpoczyna się zaraz po tej spacji, to może być również oznaczony gwiazdką

(\*), natomiast każdy znak po poprawnym rozkazie (i operandzie) jest automatycznie uznawany za komentarz i nie trzeba go specjalnie oznaczać.

## FUNKCJE REDAKCYJNE

Każdy wiersz wpisany bez numeru jest traktowany jako wiersz poleceń do natychmiastowego wykonania. Polecenia edytora **MAC/65** umożliwiają przeprowadzenie licznych operacji, które dla większej czytelności podzieliłem na dwie grupy. Jeśli polecenie zostanie podane błędnie, edytor sygnalizuje to sygnałem akustycznym i napisem **WHAT?** (co?).

Edytor umożliwia automatyczne numerowanie wierszy pisanego programu. Uruchomienie tej funkcji następuje po podaniu polecenia **NUM** (Number) z dwiema liczbami rozdzielonymi przecinkiem. Pierwsza wartość określa numer pierwszego wiersza, a druga krok między numerami. Brak drugiego parametru powoduje zwiększanie numerów wierszy o 10, zaś pominięcie obu nadaje pierwszemu numerowi pisanego wiersza wartość większą o 10 od najwyższego numeru wiersza znajdującego się w pamięci. Przerwanie automatycznej numeracji następuje po naciśnięciu klawisza **BREAK** lub gdy wiersz o numerze, który powinien nastąpić, znajduje się już w pamięci.

Znajdujący się w pamięci tekst programu może być przenumerowany poleceniem **REN** (RENumber). Dwie następujące po nim liczby mają znaczenie takie, jak w poleceniu **NUM**. Brak liczb spowoduje przenumerowanie programu tak, że pierwszy wiersz otrzyma numer 10 i kolejne numery będą wzrastały o 10.

Część programu można skasować podając polecenie **DEL** (DELeTe) z numerami pierwszego i ostatniego kasowanego wiersza. Pierwszy kasowany wiersz musi znajdować się w pamięci, a wartość ostatniego może być dowolna (w dozwolonym zakresie).

Dwie bardzo użytecznymi funkcjami edytora są wyszukiwanie i wymiana. W celu znalezienia w programie dowolnego tekstu podajemy polecenie **FIND**, a następnie poszukiwany ciąg ograniczony dowolnymi znakami oprócz spacji. Jeśli poszukiwany ciąg jest tylko fragmentem programu, to należy podać numer pierwszego i ewentualnie ostatniego wiersza, w przeciwnym przypadku zostanie przeszukany cały program. Pierwszy wiersz zawierający poszukiwany ciąg jest wyświetlany na ekranie. Jeśli chcemy znaleźć wszystkie takie ciągi, to na końcu polecenia należy dodać literę **A** (All) po przecinku.

Podobna jest składnia polecenia **REP** (REPlace), które realizuje zamianę ciągu. Należy jednak podać dwa ciągi — stary i nowy. Ponadto możliwa jest selektywna wymiana: po dopisaniu na końcu polecenia litery **Q** (Question) każdy znaleziony stary ciąg jest wymieniany na nowy po wpisaniu **Y** i **RETURN**, a samo **RETURN** pozostawia go bez zmiany. Jednoczesne użycie **A** i **Q** jest niedozwolone, gdyż **Q** powoduje automatycznie przeszukiwanie całego wyznaczonego obszaru.

Ponieważ nie każdy użytkownik musi biegle przeliczać między systemem dziesiętnym i szesnastkowym, to program posiada polecenie **?**, które dokonuje tej konwersji, podana po nim liczba dziesiętna zostanie zamieniona na szesnastkową i odwrotnie. Liczba szesnastkowa musi być poprzedzona znakiem **\$**.

Rozmiar użytej przez program pamięci można określić poleceniem **SIZE**. Odpowiedzią jest najniższy i najwyższy adres obszaru zajmowanego przez program użytkownika. Jeśli dolna część pamięci jest potrzebna na procedury, to dolną granicę obszaru programu przesuwamy poleceniem **LO-MEM** z nowym adresem. Polecenie to kasuje program zawarty w pamięci.

Aktualna zawartość poszczególnych komórek pamięci jest wyświetlana w odpowiedzi na polecenie **D** (Display), po którym następuje adres pierwszej komórki oraz ewentualnie adres ostatniej. Gdy adres ostatniej komórki pamięci nie zostanie podany, to wyświetlana jest zawartość ośmiu kolejnych komórek. Do zmiany zawartości tych komórek służy polecenie **C** (Change) z adresem początkowym zmian i znakiem <. Następnie należy podać nowe wartości oddzielone przecinkami. Pominięcie wartości między

# INSTALACJA MODEMU

Od kilku miesięcy są w Polsce modemy XM-301P przeznaczone dla komputerów Atari XL/XE. Wielu potencjalnych użytkowników sądzi zapewne, że zakup jest sprawą prostą, lecz jak potem to urządzenie zainstalować. Wbrew pozorom jest to operacja bardzo prosta, choć trwa od dwóch tygodni do miesiąca.

Wraz z zakupionym modemem nabywca otrzymuje polską wersję instrukcji obsługi. Zawiera ona na końcu gotowe podanie do Dyrekcji Generalnej PPTT (Polska Poczta, Telegraf i Telefon). Należy tylko wpisać swoje imię, nazwisko, adres i numer telefonu, włożyć wypełnione podanie do koperty i cierpliwie czekać na odpowiedź. Z naszej

redakcji, która mieści się przy ulicy Wspólnej, jest do Dyrekcji PPTT (pl. Małachowskiego) parę kilometrów. Odpowiedź nadeszła więc już po dwóch... tygodniach. Mam nadzieję, że na przykład do Przemyśla lub Szczecina dotrze przed końcem XX wieku. Odpowiedź ta brzmi następująco:

W odpowiedzi na Wasze pismo z dnia... Zarząd Służby Telekomunikacyjnej wyraża zgodę na przyznanie abonamentu teleinformatycznego i podłączenie modemu XM-301P do stacji telefonicznej nr...

Jednocześnie informujemy, że w sprawie abonamentu oraz modemu należy zwrócić się do Głównego Urzędu Telekomunikacji Międzygospodarczej w Warszawie.

Zwróciliśmy się oczywiście natychmiast i co? Abonament będzie naliczany automatycznie od początku następnego miesiąca, modem można już używać. A instalacja?

Technicznie problem jest bardzo skomplikowany — tak jak założenie wtyczki sieciowej na przewód — i może to zrobić każdy, kto choć raz w życiu używał śrubokrętu. Należy jedynie nabyć gniazdko telefoniczne. Można także na czas pracy modemu wyłączać aparat telefoniczny i wtedy nie trzeba nic. Zastosowałem w domu wariant pierwszy. A do redakcji wezwaliśmy fachowca z GUTM. Chyba późnie i ślizga się po śniegu, bo wymieniona instytucja znajduje

się 200 (dwieście) metrów od redakcji, a trwa to już trzy tygodnie i fachowiec jeszcze nie dotarł. Ponieważ cykl wydawniczy „Bajka” ma trzy miesiące, to jest nadzieja, że w chwili, gdy czytasz te słowa, modem już działa.

W zasadzie to już koniec, ale nasunęła mi się pewna refleksja. Rozumiem, że napisanie cytowanego wcześniej pisma kosztuje 300 złotych (dla osoby prywatnej — instytucja płaci 27 tysięcy: nawiasem mówiąc pis-

Lp.	Imię, nazwisko, nazwa	Miasto	Telefon	Uwagi
1	Maciej Rybicki	Warszawa	46-46-37	
2	Tomasz Wiśniewski	Warszawa	44-16-12	
3	Klub Komputerowy NOT	Warszawa	47-70-94	wt, czw, pt
4	NN	Zakopane	21-75	
5	Zas. Szkoła Zawod.	Gorlice	28-701	
6	Bogdan Haehne	Rzeszów	47-861	



przecinkami spowoduje pozostawienie odpowiedniej komórki pamięci bez zmian.

## ZAPIS I ODCZYT

Druga grupa funkcji **MAC/65** służy do przeprowadzania operacji zapisu i odczytu, zarówno programu źródłowego, jak i wynikowego. Program źródłowy może być przy tym zapisywany dwoma sposobami (podobnie jak w Basicu): w postaci stokenizowanej i w postaci pliku ASCII.

Zapis postaci stokenizowanej jest wykonywany na polecenie **SAVE**, po którym następuje specyfikacja pliku poprzedzona znakiem #. Np. dla magnetofonu będzie to **SAVE #C:**. Odczyt tak zapisanego programu źródłowego wykonuje polecenie **LOAD** ze specyfikacją pliku. Odczyt powoduje skasowanie aktualnej zawartości pamięci, chyba że na końcu polecenia dodamy literę **A** (Append), dzięki czemu odczytywany plik zostanie dopisany do programu aktualnie znajdującego się w pamięci.

Polecenie **LIST** powoduje listowanie programu źródłowego i ma działanie identyczne jak w Basicu. **LIST** bez specyfikacji pliku listuje program na ekran, a opuszczenie numerów pierwszego i ostatniego wiersza daje listing całego programu. Drugim poleceniem o podobnym działaniu jest **PRINT**. Różnica między **LIST** i **PRINT** polega na tym, że to ostatnie polecenie daje listing bez numerów wierszy. Program zapisany w pliku ASCII (niestokenizowanym) odczytujemy poleceniem **ENTER** ze specyfikacją pliku. Inaczej niż w Basicu, następuje tu skasowanie zawartości pamięci przed odczytaniem wskazanego pliku. Aby tego uniknąć należy na końcu polecenia dopisać **M** (Merge). Jeśli plik nie posiada numerowanych wierszy (np. był zapisany przez **PRINT**), to po dodaniu na końcu polecenia litery **A** (Autonumber) wprowadzane wiersze są automatycznie numerowane. Użycie obu wariantów (**M** i **A**) jednocześnie jest niedozwolone.

Zawartość dowolnego obszaru pamięci, np. zawierającego kod wynikowy, może być zapisana przy pomocy polecenia **BSAVE**, po którym musi nastąpić specyfikacja pliku (ze znakiem #), znak < oraz adres pomocniczy i końcowy zapisywanego obszaru. Polecenie **BSAVE** jest odpowiednikiem funkcji Binary Save DOS-u. Analogicznie, odpowiednikiem funkcji Binary Load jest polecenie **BLOAD**, które odczytuje wskazany plik binarny do pamięci. Przy odczycie plik jest umieszczany w pamięci w miejscu wyznaczonym przez adresy zawarte w jego nagłówku.

## OPERATORY

Jak każdy język programowania, także assembler musi zawierać zestaw operatorów umożliwiających wykonywanie pewnych działań. W **MAC/65** są to operatory podstawowych działań arytmetycznych (+, -, \*, /), operatory relacji (=, >, <, >=, <= i <>), operatory logiczne (**OR**, **AND** i **NOT**) oraz operatory operacji binarnych **&** — **AND**, **I** — **OR** i **^** — **EOR**. Operatory relacji i operatory logiczne dają zawsze wynik prawda (1) lub fałsz (0), natomiast operatory arytmetyczne i binarne działają na liczbach 16-bitowych zgodnie z zasadami obowiązującymi dla odpowiednich rozkazów procesora 6502, lecz bez tworzenia bitu przeniesienia (Carry). Ponadto znaki > i < służą do oznaczenia odpowiednio starszego i młodszego bajtu wyrażenia. Ponieważ zwykłe nawiasy okrągłe są zastrzeżone dla oznaczenia trybów adresowania procesora 6502, to w wyrażeniach zamiast nich należy stosować nawiasy kwadratowe ([ i ]).

Operator równości służy jeszcze do nadawania wartości etykietom i określenia miejsca ulokowania kodu wynikowego. Umieszczenie po etykiecie znaku = i liczby lub wyrażenia spowoduje nadanie tej etykiecie stałej wartości. Wartość ta nie może zostać zmieniona w żadnym miejscu programu. Jeśli jednak operator równości poprzedzimy kropką (.=), to tak określona wartość etykiety może zostać zmieniona w dowolnym miejscu programu.

Położenie kodu wynikowego jest oznaczane w każdym miejscu programu przez gwiazdkę (\*). Symbolizuje ona tzw. licznik lokacji. Na początku programu należy nadać mu odpowiednią wartość przy użyciu operatora = (gwiazdka musi być tu oddzielona od numeru wiersza etykiety lub co najmniej dwoma spacjami). Jeśli wewnątrz progra-

mu trzeba zarezerwować kilka bajtów na dane, to licznik lokacji może być użyty w znanej z Basicu instrukcji przypisania, np. **\*=+4**. Podobnie można oznaczać skoki względne, np. **BNE \*-12**.

## DYREKTYWY

Do poprawnego zapisu programu konieczne są jeszcze tzw. dyrektywy assemblera. Są to polecenia wykonania pewnych operacji, których rezultatem może być również wygenerowanie kodu wynikowego w odpowiedniej formie.

Do tworzenia ciągu bajtów służy dyrektywa **.BYTE**. Po niej należy wpisać odpowiednie wartości w formie liczb, wyrażen lub znaków. Pojedyncze znaki są przy tym oznaczane apostrofem ('), a ciągi znaków muszą być ujęte w cudzysłowy. Wszystkie rodzaje danych można dowolnie mieszać w jednej dyrektywie, np.

**.BYTE 22 + START, 'Y', "Start", \$9B**

Bardzo podobne działanie mają dwie następne dyrektywy. **.CBYTE** tworzy również ciąg bajtów, lecz wartość ostatniego bajtu jest związana o 128 (\$80). Np. dyrektywa **.CBYTE "AAA"** spowoduje utworzenie bajtów \$40, \$40, \$C0. Druga dyrektywa — **.SBYTE** — generuje bajty w kodzie ekranowym (wewnętrznym) komputera Atari. Odpowiadają one wartościom, które trzeba użyć w instrukcji POKE w celu umieszczenia znaków na ekranie.

Powyższe dyrektywy operują na wartościach jednobajtowych. Dla uzyskania wartości dwubajtowych służą dyrektywy **.WORD** i **.DBYTE**. Pierwsza z nich tworzy kod w normalnym formacie 6502, czyli w kolejności młodszy bajt, starszy bajt, zaś druga odwrotnie (starszy, młodszy).

Przy wykonywaniu operacji na liczbach rzeczywistych bardzo pomocna jest dyrektywa **.FLOAT**, która generuje sześciobajtowy kod podanej wartości w stosowanym przez Atari formacie zapisu liczb zmiennoprzecinkowych. Np. **.FLOAT 1** utworzy ciąg bajtów: \$40, \$01, \$00, \$00, \$00 i \$00.

Przy opisie etykiet wspominałem, że etykiety rozpoczynają się od znaku ? są traktowane jako lokane, czyli są rozpoznawane tylko w wyznaczonym (lokalnym) obszarze programu. Normalnie jest to obszar od początku do końca programu. Dyrektywa **.LOCAL** zamyka jeden obszar lokalny i otwiera następny. Wszystkie etykiety lokalne zawarte między dwoma dyrektywami **.LOCAL** będą „niewidoczne” dla kompilatora poza obszarem ograniczonym przez te dyrektywy. Umożliwia to stosowanie w kilku miejscach programu jednakowych nazw dla etykiet o różnych wartościach.

Kolejne dyrektywy służą do ustalenia parametrów pracy assemblera. Dyrektywa **.TAB** określa kolumny ekranu, w których rozpoczyna się listowanie rozkazów, operandów i komentarzy. Standardowym ustawieniem jest **.TAB 8,12,28**. Druga dyrektywa — **.SET** — pozwala na określenie parametrów listingu. Należy podać po niej numer parametru i jego żadaną wartość. Kolejnymi parametrami są:

- 0 — liczba bajtów w polu kodu dla dyrektyw **.BYTE** i **.SBYTE** (od 1 do 4, normalnie 4);
- 1 — lewy margines listingu (0—31, 0);
- 2 — długość wiersza dla drukarki (40—132, 80);
- 3 — odstęp między wierszami na drukarce (0 lub 12, 12) — wartość 0 trzeba podać, gdy drukarka sama przesuwu papier po zakończeniu wiersza;
- 4 — długość strony w wierszach (dowolna, 66).

## ASEMBLACJA

Gotowy program trzeba skompilować, czyli przekształcić na kod maszynowy. Operacja ta jest nazywana asemlowaniem i wykonywana na polecenia **ASM** (ASseMbly). W poleceniu mogą być określone pliki, które mają być użyte w tej operacji. Kolejno są to: plik źródłowy (z programem do kompilacji), plik listingu (z listingiem kompilacji), plik wynikowy (z programem wynikowym) i plik etykiet (tymczasowy plik zawierający adresy etykiet). Jeśli jeden z parametrów zostanie pominięty, to należy po prostu nie wpisywać specyfikacji pliku między przecinkami.

Opuszczenie któregoś z parametrów asemlowania spowoduje przyjęcie przez kompilator wartości standardowej. Przy braku pliku źródłowego kompilowany jest

program zawarty w pamięci. Brak pliku listingu wyświetla listing asemlacji na ekranie. Jeśli nie podamy nazwy pliku wynikowego, to kod wynikowy nie będzie tworzony w ogóle lub zostanie umieszczony w pamięci (zob. dalej). Wykaz etykiet jest także przechowywany w pamięci, jeżeli nie przewidzieliśmy dla niego specjalnego pliku.

Pliki, których używamy przy asemlowaniu programu, mają ustalone formaty. Plik źródłowy musi być zapisany poleceniem **SAVE**, plik wynikowy otrzyma format pliku binarnego DOS, a plik listingu będzie zwykłym plikiem tekstowym ASCII.

Przebieg asemlowania może być ustalony przez dyrektywę. **.OPT** umieszczoną w programie. Występuje ona w dwóch wariantach: **.OPT** uruchamia opcję assemblera, a **.OPT NO** wyłącza ją. Dalszym parametrem dyrektywy jest nazwa opcji. Są one następujące:

**LIST** — cały listing asemlacji lub (NO LIST) tylko listing wierszy zawierających błędy;

**ERR** — wskazywanie błędów występujących przy asemlowaniu;

**OBJ** — tworzenie kodu wynikowego, przy asemlowaniu do pamięci kod wynikowy jest tworzony TYLKO po podaniu dyrektywy **.OPT OBJ**, zaś przy asemlowaniu do pliku jest tworzony ZAWSZE, gdy nie było dyrektywy **.OPT NO OBJ**;

**EJECT** — umieszczenie w listingu numeru strony i tytułu.

Tytuł listingu, drukowany lub wyświetlany na początku każdej strony obok jej numeru, jest określany przez użytkownika przy pomocy dyrektywy **.TITLE**. Tytułem będzie ciąg znaków znajdujących się po tej dyrektywie (maksymalnie 70 znaków). Ciąg ten nie musi być umieszczany w cudzysłowach. Podobne działanie ma dyrektywa **.PAGE**, która powoduje przejście do początku następnej strony i umieszczenie pod numerem strony i tytułem listingu nagłówka podanego jako ciąg w dyrektywie.

Program źródłowy zajmuje zawsze więcej miejsca w pamięci niż program wynikowy. Powoduje to ograniczenie wielkości tworzonych programów w języku maszynowym. Na ominięcie tej niedogodności pozwala dyrektywa **.INCLUDE**. Dołącza ona do asemlowanego programu program źródłowy z urządzenia zewnętrznego. Program taki musi być zapisany w postaci stokenizowanej (przez **SAVE**). Ponadto dzięki niej możliwe jest włączenie do programu znajdujących się na dyskiecie **MAC/65** plików **SYSEQU.M65** i **IOMAC.LIB**. Pierwszy z nich zawiera systemowe etykiety Atari, a drugi bibliotekę standardowych operacji wejścia/wyjścia.

Poza opisanymi wyżej funkcjami **MAC/65** pozwala również na stosowanie makrorozkazów i asemlacji warunkowej. Ponieważ funkcje te należą już do zaawansowanych technik programowania, to na razie zostaną pominięte. Wróćmy do nich w końcowej części naszego kursu.

Wojciech Zientara

## SKŁADNIA POLECEŃ MAC/65

ASM [#plik\_źr] [,#plik\_list] [,#plik\_wyn] [,#plik\_etyk]  
 BLOAD #spec\_pliku  
 BSAVE #spec\_pliku <adr\_pocz, adr\_końc  
 C adres\_pocz <,(, (liczba) [(, (liczba)...]  
 D adres\_pocz [,adres\_końc]  
 DEL num\_pocz [,num\_końc]  
 ENTER #spec\_pliku [(,M) (,A)]  
 FIND (ciąg) [num\_pocz [,num\_końc]] [,A]  
 LIST [#spec\_pliku] [,num\_pocz [,num\_końc]]  
 LOAD #spec\_pliku [,A]  
 NUM [num\_pocz [,krok]]  
 PRINT [#spec\_pliku] [,num\_pocz [,num\_końc]]  
 REN [num\_pocz [,krok]]  
 REP (stary\_ciąg/nowy\_ciąg) [num\_pocz [,num\_końc]]  
 [(,A) (,Q)]  
 SAVE #spec\_pliku

mo powinno być na papierze czerpanym i ręcznie pisane złotym atramentem). Dlaczego jednak należy miesięcznie dopłacać po 1500 złotych (instytucje 4500)? Przecież i tak za każde połączenie trzeba płacić według licznika, a żadnej dodatkowej usługi poczta nie świadczy. Podobnie jest zresztą z radiotelefonami, których właściciele muszą płacić pocztą grube pieniądze za korzystanie z... eteru, którego nawet nie ma. Rzecznik praw obywatelskich ma wystarczająco dużo pracy i nie śmiem zwracać głowy takimi drobiazgami, ale może jakaś kompetentna osoba wyjaśniłaby ten problem. Takie postępowanie skutecznie może rozłożyć nie tylko komputeryzację, ale wszelki postęp techniczny w Polsce.

Wojciech Zientara

Zwracamy się z prośbą do wszystkich posiadaczy modemów o przesłanie do naszej redakcji numeru swojego telefonu. Obok publikujemy telefony pierwszych posiadaczy modemów, którzy się do nas zgłosili.

**Doczekaliśmy się kolejnego listu czytelnika adresowanego do naszej rubryki. Tym razem napisał Przemysław Kucharzewski, uczeń II klasy LO ze Świebodzina. Oto jego trzy poprawki:**

### CRYSTAL RIDERS

Liczba "życ" w komórce \$3B9F. Zamieniamy DEC \$3B9F (\$CE,9F,3B; "N" w neg., "CTRL-\*" w neg., ";") na LDA \$3B9F — pierwszy bajt (\$CE; "N" w neg.) na \$AD; "-" w neg. (dwa kolejne bez zmian).

### FROGGIE

Zamieniamy DEC \$B8 (\$C6, B8; "F" w neg., "8" w neg.) na LDA \$B8 — pierwszy bajt na \$A5; "%" w neg.

### BLAST

Liczba statków w komórce \$2C1A. Zamieniamy DEC \$2C1A (\$CE,1A,2C; "N" w neg., "CTRL-Z" w neg., ";") na LDA \$2C1A — pierwszy bajt na \$AD; "-" w neg.

W sześciu kolejnych grach poprawka polega na zamianie pierwszego bajtu w ciągu (\$CE; "N" w neg.) na \$AD; "-" w neg.

### BREW BIZ

DEC \$32E1 (\$CE,E1,32; "N" w neg., "a" w

neg., "2") — występuje w programie dwukrotnie  
 DEC \$32E2 (\$CE,E2,32; "N" w neg., "b" w neg., "2")

### OLLIES FOOLIES

DEC \$3681 (\$CE,81,36; "N" w neg., "CTRL-A" w neg., "6")

DEC \$3680 (\$CE,80,36; "N" w neg., "CTRL-" w neg., "6")

### GHOST CHASER

DEC \$27D3 (\$CE,D3,27; "N" w neg., "S" w neg., "SHIFT-7") — występuje w programie dwukrotnie

DEC \$27D4 (\$CE,D4,27; "N" w neg., "T" w neg., "SHIFT-7")

### LEAPER

DEC \$1B9F (\$CE,9F,1B; "N" w neg., "CTRL-\*" w neg., "ESC")

### ZORRO (I i II)

I — DEC \$0625 (\$CE,25,06; "N" w neg., "SHIFT-5", "CTRL-F")

II — DEC \$0498 (\$CE,98,04; "N" w neg., "CTRL-X" w neg., "CTRL-D")

### TRAILBLAZER

DEC \$1D27 (\$CE,27,1D; "N" w neg., "SHIFT-7", "CTRL-=")

DEC \$1D28 (\$CE,28,1D; "N" w neg., "SHIFT-9", "CTRL-=")

Dwie następne poprawki polegają na zamianie pierwszego bajtu w ciągu (\$C6; "F" w neg.) na \$A5; "%" w neg.

### PRELIMINARY MONTY

DEC \$88 (\$C6,88; "F" w neg., "CTRL-H" w neg.)

### CLOWNS & BALOONS

DEC \$EA (\$C6, EA; "F" w neg., "j" w neg.) — występuje w programie dwukrotnie  
 DEC \$EB (\$C6,EB; "F" w neg., "k" w neg.)

### GREMLINS

DEC \$53 (\$C6,53; "F" w neg., "S") — zamieniamy pierwszy bajt na \$A5; "%" w neg.  
 DEC \$0B9B,X (\$DE,9B,OB; "SHIFT-\*" w neg., "ESC", "CTRL-K") — zamieniamy pierwszy bajt na \$BD; "=" w neg.

### MR ROBOT

JMP \$8CF4 (\$4C,F4,8C; "L", "t" w neg., "CTRL-L" w neg.) — zamieniamy pierwszy bajt na \$5C; "SHIFT-+"

### HERBERT

DEC \$5086,X (\$DE,86,50; "SHIFT-\*" w neg., "CTRL-F" w neg., "P") — zamieniamy pierwszy bajt na \$BD; "=" w neg.

P.S. Do rubryki „Zostań nieśmiertelnym!” w „Bajtku” nr 11/88 zakradł się błąd. Poprawka do gry SYNTROON powinna wyglądać następująco:  
 ROZKAZ: DEC \$9D (\$C6,9D — "F" w neg., "ESC-CTRL-=" w neg.) należy zamienić na:  
 LDA \$9D (\$A5,9D — "%" w neg., "ESC-CTRL-=").

Tomasz Wiśniewski



# KLAN COMMODORE

# MONITORY ML

## cz IV

**Ponieważ w poprzednim odcinku zakończyliśmy prezentację standardowych poleceń monitorów warto byłoby chyba przeciwzyć nabyte wiadomości praktycznie. W tym celu napiszemy krótki program w języku wewnętrznym i za pomocą monitora wykonamy na nim szereg operacji:**

```
2710 CLC
2711 LDA #02
2713 ADC #02
2715 STA $2719
2718 BRK
```

Zadaniem tego programu jest dodanie do siebie dwóch liczb — 2 i 2. Wynik operacji zostanie zapisany w komórce pamięci o adresie \$2719. Zanim jednak wpiszesz ten program do pamięci wykonaj:

```
F 2710 2720 00
Operacja ta jak pamiętasz z poprzedniej części ma za zadanie wpisanie do wszystkich komórek od adresu $2710 do $2720 samych zer. Aby sprawdzić czy istotnie monitor wykonał dane mu polecenie możesz wykonać:
```

```
M 2710 2720
lub
I 2710 2720
w zależności od posiadanego przez Ciebie programu.
Teraz wpisz podany powyżej program i uruchom go za pomocą:
G 2710
Na ekranie zobaczysz znany Ci już komunikat ukazujący się na ekranie w chwili uruchomienia monitora.
```

Zadanie jakie realizuje program jest bardzo proste. Pierwsze polecenie (CLC) powoduje wyzerowanie znacznika przeniesienia w rejestrze słowa stanu mikroprocesora (będziemy o nim mówić w następnym artykule). Następnie do rejestru akumulatora wczytywana jest liczba 2. ADC # \$02 nakazuje dodanie do liczby zapisanej w akumulatorze następnej wartości — także 2. Po wykonaniu dodawania zapisujemy wynik (zawarty w tym momencie w akumulatorze) w komórce o adresie \$2719. BRK pozwala nam przerwać wykonywanie programu i powrócić do monitora.

Z powyższego omówienia wynika więc, że suma powinna być zapisana w komórce \$2719. Sprawdźmy czy tak jest w istocie:

```
M 2710 2720
Jak widać ostatnią wartością różną od zera w wyświetlanym ciągu jest właśnie liczba 4 czyli wynik wykonanej przed chwilą operacji. Zbadajmy teraz zawartość rejestrów mikroprocesora aby przekonać się jaka liczba znajduje się w akumulatorze:
```

R  
Zwróć także uwagę, że akumulator (AC) zawiera liczbę 4 czyli wynik naszej operacji.

Podczas omawiania tego przykładu popełniliśmy jednak pewne znaczące odstępstwo. Pamiętaj, że dobry program wie o tym, że nie jest nieomylny i ZAWSZE zapisuje świeżo wpisany program na dyskietce czy kasie PRZED jego uruchomieniem. Jeżeli zależy Ci jednak aby swoje programy wpisywać po kilkanaście razy możesz oczywiście z takiego profilaktycznego zapisu zrezygnować...

Operację zapisu przeprowadzimy następująco:  
S „DODAWANIE”, 08, 2710, 2719  
lub  
S „DODAWANIE”, 01, 2710, 2719  
jeżeli dysponujesz magnetofonem. W wypadku błędu może okazać się, że Twój mo-

onitor nie toleruje zera umieszczonego przed numerem urządzenia (wpisz wtedy odpowiednio 1 lub 8).

Jednym z najbardziej powszechnych błędów zapisu jest pomijanie ostatniego bajtu programu. W naszym przykładzie drugi adres (2719) absolutnie nie oznacza, że chcę zapisać wraz z programem wynik operacji; wszystkie monitory wymagają podawania zawsze adresu końca programu o jeden większego niż koniec rzeczywisty. Dzieje się tak dlatego, ponieważ system operacyjny komputera wymaga podania informacji nie o KOŃCU programu ale o PIERWSZYM wolnym bajcie pamięci ZA programem. Jeżeli jeszcze nie dowierzasz to wykonaj następujące operacje:

```
s „DODAWANIE”, 08, 2710, 2718
F 2710 2720 FF
L „DODAWANIE”, 08
D 2710 2720
```

Ostatnia operacja uwidoczni Ci, że dziwnym trafem ostatnia instrukcja programu (BRK — 0 dziesiętnie) gdzieś znikła...

Załóżmy teraz, że wpisany program chciałbyś przenieść czy w jakiś sposób zmodyfikować aby wykonywał on następującą operację:

```
2+2+5
W tym wypadku istnieją dwie metody rozwiązania problemu. Pierwsza z nich (bardziej żmudna) polega na przepisaniu programu prawie od nowa i dodaniu doń nowej instrukcji — ADC #05 bezpośrednio po ADC. #02. Druga — prostsza — to przesunięcie części programu o dwa bajty (jeden na mnemonik ADC, drugi na operand #05) w „górze” pamięci:
```

```
T 2715 2718 2717
```

Teraz dopisz:

```
A 2715 ADC #05
```

lub popraw poprzedni zapis po wyświetleniu programu na ekranie (za pomocą polecenia D). Następnie wykonaj:

```
D 2710 271A
a na ekranie ukaże Ci się nowa wersja programu. Efektem działania programu będzie wynik = 9, lecz wstrzymaj się jeszcze z uruchomieniem programu!
```

Spróbujmy teraz „oszukać” monitor i sprawdzić działanie instrukcji C. W tym celu przeniesiemy najpierw nasz program w inny obszar pamięci:

```
T 2710 271A 2720
```

a następnie np. za pomocą instrukcji I lub M zmienimy w poleceniu ADC s05 operand na s04. Porównanie obu wersji wykonujemy za pomocą:

```
C 2710 271A 2720
```

Na ekranie ujrzysz:

```
2716
```

co jest sygnałem, że w porównywanych programach nie zgadzają się ze sobą wartości komórek \$2716 i \$2726. Po bliższym przyjrzeniu się obu wersjom przekonasz się, że jedną z nich jest komórka zawierająca zmieniony przed chwilą operand \$04...

Przed wykonaniem programu musimy jednak wprowadzić do niego pewną zmianę. Otóż po dodaniu nowego rozkazu (ADC #05) uległy przesunięciu (o dwa bajty) instrukcje STA \$2719 oraz BRK. Gdybyśmy pozostawili nasz program w takim stanie i usiłowali go uruchomić, to komputer prawdopodobnie zablokowałby się; powodem jest tu adres \$2719 poprzednio wolny lecz teraz zajęty przez starszy bajt adresu ... \$2719 !. Należy więc teraz zmienić adres komórki, w której zapiszemy wynik operacji np. \$271B. Nowa wersja programu będzie więc wyglądała następująco:

```
2710 CLC
2711 LDA s02
2713 ADC s02
2715 ADC s05
2717 STA s271B
271A BRK
```

Jeżeli dysponujesz komputerem Commodore 128 to powyższy program możesz

uruchomić nie tylko za pomocą G 2710; również dobrze można tego dokonać za pomocą polecenia J 2710.

Pozostawmy na chwilę przykłady (wróćmy do nich niebawem) i zajmijmy się listą rozkazów mikroprocesorów stosowanych w komputerach Commodore. Ich zestawienie alfabetyczne mówi zapewne niewiele początkującym Czytelnikom: spróbujmy zatem pogrupować je w zależności od ich funkcji:

### 1. ROZKAZY LOGICZNE: AND, EOR, ORA

Te trzy rozkazy umożliwiają wykonywanie operacji logicznych AND, EXCLUSIVE OR oraz OR na liczbach zapisanych w akumulatorze bądź w konkretnych komórkach pamięci.

### 2. WARUNKOWE ROZKAZY SKOKÓW: BCC, BCS, BEQ, BMI, BNE, BPL, BYC, BYS

Są to odpowiedniki instrukcji IF...THEN w BASIC. Zajmiemy się nimi bliżej po omówieniu rejestru słowa stanu mikroprocesora.

### 3. ROZKAZY ODNOSZĄCE SIĘ DO ZMIANY STANU LOGICZNEGO BITÓW W REJESTRZE SŁOWA STANU: CLC, CLD, CLI, CLY, SEC, SED, SEI

Znaczniki rejestru słowa stanu umożliwiają mikroprocesorowi rozeznanie się jaki był wynik określonej operacji; to dzięki nim jesteśmy w stanie wykonywać np. skoki warunkowe, włączać i wyłączać dziesiętny tryb pracy procesora, przerwania IRQ itp.

### 4. PORÓWNIANIA: CMP, CPX, CPY, BIT

Polecenia te umożliwiają nam określenie czy liczba w danym rejestrze jest równa określonej wartości (CMP, CPX i CPY) lub czy dany bit (bity) w akumulatorze i określonej komórce pamięci mają ten sam stan logiczny (BIT).

### 5. OPERACJE NA BITACH: ASL, LSR, ROL, ROR

Rozkazy te umożliwiają nam przesuwanie układu bitów w lewo lub w prawo i co za tym idzie przeprowadzanie prostych operacji dzielenia i mnożenia (ROL i ROR).

### 6. ZWIĘKSZANIE I ZMNIEJSZANIE ZAWARTOŚCI REJESTRÓW: DEC, DEX, DEY, INC, INX, INY

Trzy pierwsze rozkazy powodują zmniejszanie aktualnej liczby zapisanej w jednym z rejestrów lub komórce pamięci z krokiem równym 1; trzy następne realizują operację przeciwną z tym samym przyrostem.

### 7. WCZYTYWANIE I ZAPISYWANIE LICZB DO/Z REJESTRÓW: LDA, LDX, LDY, STA, STX, STY

### 8. WYMIANA PARAMETRÓW MIĘDZY REJESTRAMI: TAX, TAY, TSX, TXA, TXS, TYA

Rozkazy te umożliwiają przenoszenie liczb zapisanych w rejestrach .A, .X i .Y oraz wartość wskaźnika stosu (TSX i TXS) pomiędzy sobą.

### 9. OPERACJE NA STOSIE MIKROPROCESORA: PHA, PHP, PLA, PLP

### 10. INSTRUKCJE SKOKÓW: JMP, JSR

### 11. POWRÓT Z PODPROGRAMU LUB PROCEDURY PRZERWANIA: RTS, RTI

### 12. ARYTMETYKA (DODAWANIE I ODEJMOWANIE): ADC, SBC

### 13. INNE ROZKAZY: NOP i BRK

Pierwszy z nich zatrzymuje pracę procesora na 2 cykle zegarowe, drugi wymusza zakończenie wykonywania programu (i powrót do monitora).

W następnym odcinku zajmijmy się omówieniem jednego z kluczowych rejestrów mikroprocesora — rejestru słowa stanu bez którego wiele operacji byłoby po prostu niemożliwe. Ze swojej strony zachęcam Czytelników do poprobowania własnych sił z monitorami — w nauce języka maszynowego duża ilość praktyki oznacza dużą sprawność programisty.

*Klaudiusz Dybowski*

# JEZYK

**Po męczącym, ale koniecznym wstępie, poświęconym innym systemom liczenia i podstawowej terminologii, jaka będzie stosowana podczas kursu, dziś przechodzimy już do pierwszych instrukcji i pierwszych programów. Te pierwsze kroki nie są dobrane tak, by pokazać Wam, że język maszynowy jest łatwy. Po prostu przerobienie i przeciwienie we własnym zakresie przedstawianych instrukcji uświadomi Wam, że wszystko jest takie proste.**

Dzisiaj zajmijmy się przesyłaniem liczb z i do pamięci oraz powiemy jak wymieniać między sobą zawartości rejestrów mikroprocesora. Jak pamiętamy rejestrami tymi są akumulator oznaczany często przez .A oraz rejestry X i Y oznaczane podobnie (.X i .Y). Wszystkie te rejestry są ośmiobitowe, a więc każdy z nich może służyć do przechowania czy manipulowania ośmioma bitami. Każdy z rejestrów możemy zapisać kopią informacji z pamięci, jak również możemy przesłać zawartość danego rejestru do zapamiętania w pamięci. Ważnym faktem wymagającym uzmysłowienia jest to, że instrukcje wczytywania i przesyłania polegają na kopiowaniu. Oznacza, że jeżeli chcemy wpisać do akumulatora (instrukcja LDA od angielskiego Load Accumulator) zawartość komórki pamięci o adresie np. \$1234 to po wykonaniu instrukcji LDA \$1234 akumulator zawierać będzie liczbę, która do tej pory była w komór-

# LISTA

**Mój problem zaczął się w momencie gdy otrzymałem w prezencie Commodore 64, drukarkę NX-1000 i stację dysków 1541. Było to moje pierwsze zetknięcie się z tajemniczym światem komputerów. Po zasięgnięciu rady u bardziej wtajemniczonych dowiedziałam się (najpierw ku uciesze), że mój komputer jest bogato w Polsce oprogramowany. Następnie przyszło jednak dość bolesne zetknięcie z rzeczywistością — gdy chciałam kupić programy, nawet sprzedawcy nie bardzo wiedzieli, do czego one służą, a tytułów mieli setki. Ponieważ moje zainteresowania się dość ukierunkowane (grafika) chciałam kupić od razu taki program, który spełni**



ce pamięci o adresie \$1234, ale nadal wspomniana komórka pamięci zawierać będzie tą samą liczbę, wpisaną przed chwilą do akumulatora. Na tym właśnie polega istota kopiowania związana z tymi instrukcjami. Podobnie oczywiście jest przy przesyłaniu informacji z np. akumulatora do komórki o adresie \$4321. Zakładając, że akumulator zawiera liczbę \$FF (pamiętamy, że znaczek \$ oznacza, iż liczba zapisana jest szesnastkowo), to po wykonaniu instrukcji przesłania STA \$4321 (od angielskiego STore Accumulator in memory) zarówno akumulator jak i komórka \$4321 zawierają liczbę \$FF. Jest to bardzo ważna cecha, gdyż można pewnym komórkom nadać na początku programu stałe wartości (tzn. takie, które nie będą się zmieniać podczas całego działania programu) i wielokrotnie odwoływać się do nich instrukcją Load nie spowoduje zmiany ich zawartości. Oczywiście musimy jednocześnie zadbać, by nie zmienić zawartości tych komórek wykonując do nich przesłanie informacji za pomocą instrukcji Store.

Ważną cechą procesorów rodziny 65xx jest fakt, że nie mają one instrukcji umożliwiającej bezpośrednią wymianę informacji między dwoma komórkami pamięci. Wszystkie tego typu operacje muszą odbywać się za pośrednictwem rejestrów .A, .X i .Y.

## Zadanie 1

Mając określone liczby w pamięci, w komórkach o adresach np. \$2800 i \$2801 chcemy zamienić między sobą zawartości tych komórek.

## Rozwiązanie

Jak już się dowiedzieliśmy nie ma instrukcji umożliwiającej wykonanie zadania pojedynczą operacją. Tak więc

musimy któryś z rejestrów zapisać zawartością komórki \$2800 i następnie przesłać zawartość tego rejestru do komórki o adresie \$2801. ALE zapomnieliśmy o komórce \$2801, której zawartość zostanie w ten sposób zniszczona. Musimy więc zrobić to trochę inaczej. Do dwóch rejestrów wczytujemy zawartości komórek \$2800 i \$2801 (np. zawartość \$2800 do rejestru .A i \$2801 do rejestru .X). Następnie zawartość .A prześlemy do komórki \$2801, a zawartość .X do \$2800. W ten sposób dokonamy wymiany informacji między tymi dwoma komórkami. Program wyglądać będzie następująco:

```
LDA $2800
LDX $2801
STA $2801
STX $2800
```

Jest to program zapisany w sposób bardziej zrozumiały dla człowieka niż dla komputera. Musimy więc przetłumaczyć ten program na postać jaką występuje w komputerze. W przyszłości gdy posługiwać się będziemy monitorami poniższe zabiegi nie będą konieczne, ale na początku warto zapoznać się z tym, jak będzie wyglądać nasz program po wpisaniu go do komputera. Każda z instrukcji posiada swój kod, np. dla instrukcji LDA adres jest nim \$AD. Poniżej przedstawiam tabelkę ze wszystkimi kodami dla instrukcji typu LD-adres i ST-adres:

```
LDA-$AD LDX-$AE LDY-$AC STA-$8D
STX-$8E STY-$8C
```

Kody te mówią komputerowi co będzie musiał za chwilę zrobić oraz podpowiadają, że po takim kodzie wystąpią ponadto dwie liczby oznaczające młodszy i starszy bajt wybranego adresu (pamiętajmy, że adresy są zapisywane w dwóch bajtach i składają się

z 16 bitów). Tak więc po kodzie AD zapisujemy 00 28 (najpierw młodszy a potem starszy bajt). Postępując podobnie dla każdej instrukcji naszego programu otrzymujemy:

```
AD 00 28 LDA $2800 wpisz do akumulatora
                        zawartość komórki $2800
AE 01 28 LDX $2801 wpisz do rejestru .X
                        zawartość komórki $2801
8D 01 28 STA $2801 prześlij zawartość akumulatora
                        do komórki $2801
8E 00 28 STX $2800 prześlij zawartość rejestru .X
                        do komórki $2800
00 BRK Koniec programu (BREAK)
```

Po wpisaniu powyższego programu do komputera i wykonaniu instrukcji M adres (patrz opis monitorów) zobaczylibyśmy co następuje AD 00 28 AE 01 28 8D 01 28 8E 00 28 00. Tak więc cały program zajął nam 13 bajtów.

Pozostał jeszcze jeden problem do rozwiązania — gdzie umieścić nasz program. Wspólnie z Klaudiuszem uzgodniliśmy, że programy będą wpisywane od adresu \$2710 (10000 dziesiętnie), gdyż jest to obszar pamięci wspólny dla całej rodziny komputerów COMMODORE, umożliwiający wpisywanie programów z tego kursu bez ich modyfikacji na poszczególne modele komputerów. Możemy się zabrać do wpisywania tego programu. Wpisujemy M 2710 2720. Otrzymujemy na ekranie obraz pamięci pomiędzy adresami \$2710 \$2720. Poczynając od pierwszego widocznego kodu wpisujemy nasze 13 bajtów kodu. Pozostałymi wartościami nie należy się przejmować. Ponieważ chcielibyśmy zobaczyć rezultat działania naszego programu wpisujemy konkretne wartości do komórek \$2800 i \$2801. Niech to będą wartości odpowiednio \$DF i \$25. Wpisujemy posługując się monitorem M 2800 2810 i zmieniamy pierwsze

dwie liczby po adresie na DF i 25. To już wszystko. Pozostało nam tylko uruchomić program. Dokonujemy tego instrukcją G 2710. Po wykonaniu się programu (momentalnie) wpisujemy M 2800 2810 i patrzymy na pierwsze dwie liczby. Tym razem powinny one wystąpić w odwrotnej kolejności tzn. 25 DF. Jeżeli tak jest to program został wpisany poprawnie a my mamy już za sobą pierwszy program w języku maszynowym. Jako zadanie domowe proponuję napisać program, który będzie wymieniać zawartości komórek \$2800 \$2801 \$2802 \$2803 \$2804 w takiej właśnie kolejności. Rozwiązanie tego zadania ukaże się za miesiąc. Pozostało nam jeszcze omówienie instrukcji związanych z przesyłaniem zawartości rejestrów pomiędzy sobą. Są to instrukcje zaczynające się od litery T a następnie podające nazwę rejestru, z którego przesyłamy wartość i nazwę rejestru, do którego przesyłamy np. TAX oznacza „prześlij zawartość akumulatora do rejestru .X” (od angielskiego Transfer Accumulator to index X). Inne kombinacje też są możliwe TXA, TYA, TAY itp. LISTING1 przedstawia nasze zadanie napisane przy użyciu właśnie tych instrukcji. Do tego czym różnią się te dwa programy i jak działa ten ostatni spróbujcie już dojść sami.

*Dominik Falkowski*

READY.

MONITOR

PC SR AC XR YR SP

B000 00 00 00 00 F8

02710 AC 00 28 LDY \$2800

02713 AE 01 28 LDX \$2801

02716 98 TYA

02717 8D 01 28 STA \$2801

0271A 8A TXA

0271B 8D 00 28 STA \$2800

0271E 00 BRK

# UŻYTKÓW

**wszystkie moje oczekiwania. Tylko skąd mam jako początkującą wiedzieć, jak się ten program nazywa... [...]"**

Podobnych listów przychodzi do nas coraz więcej czemu nie bardzo się dziwimy gdyż rzeczywiście Commodore jest jednym z najlepiej oprogramowanych komputerów w Polsce. Dlatego postanowiliśmy zrobić konkurencję LIŚCIE PRZEBOJÓW wprowadzając LISTĘ UŻYTKÓW. Będziemy w niej przedstawiali programy użytkowe, które zdaniem Czytelników są najlepsze w danej dziedzinie zastosowań komputera, przy czym NIE MUSZĄ TO BYĆ programy wyłącznie firmowe.

W odróżnieniu od LISTY PRZEBOJÓW chcielibyśmy w miarę możliwości publikować także Wasze opinie na temat tych programów oczywiście w bardzo skrótovej formie. Mamy na-

dzieję, że dzięki temu Czytelnicy BAJTKA będą mieli większe rozeznanie o dostępnych w kraju programach, będą wiedzieli czego szukać na rynku.

Zasada głosowania jest taka sama jak w LIŚCIE PRZEBOJÓW z tym, że głosujemy tylko na jeden program z każdej kategorii i podajemy również informację o posiadanym zestawie komputerowym. Gdy proponowany przez Czytelników program będzie wchodził na naszą listę będziemy również podawać jego krótką charakterystykę, wymagania sprzętowe itp. Literki przy tytułach programów oznaczają niezbędne wyposażenie — stację dysków (D), magnetofon (T), drukarkę (P), monitor kolorowy (K), joystick (J), myszkę (A), modem (M), pióro świetlne (L), wiosełka (W). Jeśli dana literka będzie podana w nawiasie oznaczać to będzie, że dane urządzenie nie jest konieczne wymagane lecz jest przydatne i ułatwia pracę z danym

TYP PROGRAMU	COMMODORE 128	OSPRZET	COMMODORE 64	OSPRZET
Kalkulacyjny	SWIFTCALC 128	D	MULTIPLAN 64	D
Baza danych	dbase II (CP/M)	D	MANAGER 64	D
Edytor tekstu	FONTMASTER 128	DP	FONTMASTER II	DP
Grafika	GRAPHIC 80	DJ	GEOPAINT	DJ(A)
Grafika 3D	CAD 3D	DJ	GIGA CAD PLUS	D
Muzyka	MUSIC MAKER 128	D	ROCK MONITOR V5.0	D
Asembler	LADS	D	CBM MACROASSEMBLER	D
System	GEOS 128	DJ(A)	GEOS V1.3	DJ(A)
Rozszerz. BASIC	METABASIC 128	D	SIMON'S BASIC	D/T
Kompilator	BASIC 128	D	BASIC 64	D
Narz. dyskowy	DOS SHELL	D	FILE COPY #	D
Kopiujący	FAST HACK'EM V4.1B	D	FAST HACK'EM V4.1B	D
Matematyka	-		MATHEMAT	D
TURBO (tasma)	-		TURBO ROM	T
TURBO (dysk)	-		SUPERDOS V2.0	D
Modul	-		Action Replay Mk.5	-
Pakiet	GEOS	DJ(A)	GEOS	DJ(A)

## \* Z ZESTAWU "SUPER KIT"

programem. Pierwszą edycję LISTY UŻYTKÓW poświęcamy Commodore 64 i C-128; mamy jednak nadzieję, że posiadacze innych typów Commodore nadeślą do nas niebawem własne propozycje. Czekamy więc na Wasze li-

sty z dopiskiem LISTA UŻYTKÓW podając jednocześnie pierwsze wybrane typy.

*Klaudiusz Dybowski  
Dominik Falkowski*



# BHP VIRUS KILLER

**Wydawać by się mogło, że wirusy komputerowe dotyczą tylko sprzętu „poważnego” — IBM, AMIGI czy Atari ST. Niestety również i pocziwe komputerki 8-bitowe są podatne na tę zarazę i choć działanie wirusów odnosi się do nich w znacznie mniejszym zakresie, to jednak ich uderzenie może być dla użytkownika dość bolesne.**

Użytkownicy programów w Polsce są moim zdaniem szczególnie narażeni na działanie nowej zarazy komputerowej szerzącej się w zastraszającym tempie. Niczym nieskrępowana wymiana oprogramowania przychodzącego głównie z zagranicy jedynie sprzyja przenoszeniu wirusów z jednej programówki do drugiej; możliwość wykrycia „chorego” oprogramowania jest prawie żadna, ponieważ nie istnieją żadne uniwersalne programy „szczepionki”, a jeśli nawet to są one chowane dość skrzętnie, gdyż na oprogramowaniu tym można oczywiście nieźle zarobić.

Poniższy program pochodzi z lutowego wydania magazynu „64'er” z 1988 roku i pozwala wykryć wirus o nazwie BHP (BAYERISCHE HACKERPOST). Oryginalna wersja programu była rozbudowana dodatkowo o opcję usuwania wirusa, którą zdecydowałem się jednak z programu usunąć, ponieważ nie miałem stuprocentowej pewności czy nie wprowadza ona wirusa na dyskietkę! Tak wypadek zdarzył się już kilkakrotnie np. w wypadku Atari i programu „zabezpieczającego” przed działaniem wirusa YDO. W omawianym wypadku program powinien usuwać „zarażone” pliki z dyskietki, tymczasem były one rozbudowywane o kilka dodatkowych bloków i stąd dla pewności usunąłem ten fragment kodu.

Program ma cztery opcje: wyszukiwanie wirusa, wyświetlanie listy zbiorów „zarażonych”, wyświetlanie katalogu dyskietki oraz koniec pracy. Po wykryciu wirusa na ekranie wyświetlany jest komunikat „WIRUS BHP W ....” wraz z nazwą „chorego” programu.

Celowo napisałem, że wirusy mają dość ograniczone pole działania w komputerach 8-bitowych. Po pierwsze posiadacze takich komputerów nie dysponują raczej dyskami twardymi, stąd też infekcja dotyczy zwykle zbiorów na jednej dyskietce (co wcale nie znaczy, że ich utrata bądź zmiana nie jest bolesna!). Przenoszenie wirusa można w prosty sposób ograniczyć poprzez działania profilaktyczne — wyłączenie komputera i stacji dysków na ok. 30 sekund jeśli pracowaliśmy z dyskietką na której podejrzewamy obecność wirusa. Dzięki temu wirus ten znajdujący się prawdopodobnie w pamięci RAM stacji lub komputera nie ma żadnych szans na zainfekowanie następnej dyskietki. Czy jednak w każdej sytuacji wirus jest bez szans?

Odpowiedź na to pytanie nie jest prosta. Dobrym przykładem może być zainfekowany kompilator BASIC; każdy poddany kompilacji program stanie się wnet roznośnikiem zarazki. Generalnie wirus może docześcić się wszędzie — do katalogu dyskietki, do tytułu programu na taśmie (zaraz za nazwą), w dowolny sektor dyskietki czy na koniec programu. Po przeniknięciu na typowy (dla Commodore) nośnik, wirus może się uaktywnić jedynie w czasie gdy taśma bądź dyskietka są aktualnie wykorzystywane. Aktywizacja ta może mieć różne formy — od „doczepiania” się do kolejnych programów (tylko w wypadku dyskietki), których spis jest zawarty w katalogu, do ataku polegającego najczęściej na całkowitym skasowaniu zawartych na dyskietce zbiorów. Tego rodzaju działanie jest niemożliwe w wypadku kasy.

Dopisanie jakiegokolwiek kodu i w jakiegokolwiek miejsce dyskietki jest jednak możliwe wyłącznie wtedy, gdy wycięcie zabezpieczające dyskietkę przed zapisem jest niezaklejone. Założmy, że mamy wirusa na dyskietce i wraz z programem został on wczytany do pamięci komputera, czy (co jest bardziej prawdopodobne) do pamięci RAM stacji dysków. Jeśli teraz rozpoczniemy pracę na dyskietce „zdrowej” zabezpieczonej przed zapisem, to nie ma możliwości jej zainfekowania — wpisanie wirusa udaremnia DOS stacji. Ponieważ pamięć ROM, w której system operacyjny stacji jest zawarty nie da się zmienić, wirus jest nieszkodliwy, chyba, że przepisuje on zawartość ROM do RAM i odpowiednio modyfikuje DOS zawarty teraz już w RAM; w tym wypadku jednak musi to być dość czuły program.

Zgola inaczej wygląda sytuacja, gdy na dyskietce zapisujemy jakieś dane. Zawarty w RAM komputera czy stacji, przeważnie bardzo krótki fragment kodu maszynowego będący naszym wirusem, ma możliwość wtargnięcia nie tylko na samą dyskietkę, ale również do każdego z zawartych na niej zbiorów. Jeśli zbiór taki przeniesiemy na inny dysk, to zarazek przeniesie się również i dalej, następuje już reakcja łańcuchowa.

Dyskietki i programy z wykrytym (lub nawet spodziewanym) wirusem powinny być natychmiast formatowane bez przenoszenia zbiorów na nich zawartych na jakąkolwiek inną dyskietkę. Przed przystąpieniem do pracy z inną dyskietką wyłącz wszystkie urządzenia ze stawu — masz wtedy gwarancję, że wirus ulegnie likwidacji jeśli nawet znajdował się w komputerze. A swoją drogą lepiej nie „wymieniać” programów pochodzących z niepewnych źródeł...

Klaudiusz Dybowski

```

B1 100 POKE 53280,0:POKE 53281,0
BB 105 CL$=CHR$(147):W$=CHR$(0):Q=144
5F 110 DIM DA(32),V1(Q),V2(Q),F1(Q),TR(Q),SE(Q),
    ,VF(Q),VT(Q),VS(Q),D1(Q),D2(Q)

DD 115 FOR I=1 TO 32:READ DA(I):NEXT
60 120 PRINT CL$"BHP VIRUS KILLER":PRINT:PRINT
E2 125 PRINT"1. WYSZUKIWANIE WIRUSA":PRINT
CD 130 PRINT"2. LISTA ZBIOROW ZARAZONYCH":PRINT
7D 135 PRINT"3. KATALOG DYSKIETKI":PRINT
79 140 PRINT"4. KONIEC PRACY":PRINT:PRINT
5C 145 GET A$:IF A$="" GOTO 145
A0 150 PRINT A$: ON VAL(A$) GOTO 160,310,330,360
50 155 GOTO 145
E3 160 GOSUB 390:IF F THEN PRINT A2$:GOSUB 375:G
    OTO 120

69 165 OPEN 1,8,15,"1":OPEN 2,8,2,"*":PRINT:PRIN
    T"PASS 1"

B1 170 TR=18:SE=1:FZ=0
A5 175 PRINT#1,"U1 2 0 18":SE:SE=SE
EC 180 GET#2,A$,B$
2E 185 TR=ASC(A$+W$):SE=ASC(B$+W$)
77 190 FOR BP=2 TO 226 STEP 32
37 195 PRINT#1,"B-P 2":BP
2E 200 GET#2,A$:IF (ASC(A$+W$)AND15)<>2 THEN NEX
    T BP:GOTO 225

53 205 FZ=FZ+1:GET#2,A$,B$:TR(FZ)=ASC(A$+W$):SE(
    FZ)=ASC(B$+W$)
8F 210 F1(FZ)="":D1(FZ)=SX:D2(FZ)=BP+1
4E 215 FOR I=1 TO 16:GET#2,A$:F1(FZ)=F1(FZ)+A$
    :NEXT

9C 220 PRINT".":NEXT
0C 225 IF TR<>0 GOTO 175
AB 230 PRINT:IF FZ=0 THEN PRINT"BRAK ZBIOROW TYP
    U PRG NA DYSKIETCE.":GOTO 285

2B 235 PRINT:PRINT"PASS 2"
F4 240 VZ=0:Z=0
19 245 Z=Z+1
A7 250 PRINT#1,"U1 2 0":TR(Z):SE(Z)
94 255 GET#2,A$,B$:FT=ASC(A$+W$):FS=ASC(B$+W$):G
    ET#2,A$,A$

22 260 FOR I=1 TO 32:GET#2,A$:A=ASC(A$+W$)
B5 265 IF A=DA(I) THEN NEXT I:GOTO 290
7C 270 I=34:NEXT I
90 275 PRINT".":IF Z<>FZ GOTO 245
6C 280 IF VZ=0 THEN PRINT:PRINT:PRINT"NIE MA WIR
    USA BHP NA TEJ DYSKIETCE."

7A 285 CLOSE 2:CLOSE 1:GOSUB 375:GOTO 120
EC 290 PRINT"WIRUS BHP W "F1(Z)" !"
F9 295 VZ=VZ+1:VF(VZ)=F1(Z):VT(VZ)=TR(Z):VS(VZ
    )=SE(Z)

43 300 V1(VZ)=D1(Z):V2(VZ)=D2(Z)
2B 305 GOTO 275
26 310 PRINT CL$:IF VZ=0 THEN PRINT"BRAK PROGRAM
    OW Z WIRUSEM.":GOSUB 375:GOTO 120

0C 315 PRINT"PROGRAMY ZARAZONE.":PRINT
36 320 FOR I=1 TO VZ:PRINT VF(I):NEXT
5B 325 GOSUB 375:GOTO 120
4E 330 GOSUB 390:IF F THEN PRINT A2$:GOSUB 375:G
    OTO 120

A9 335 PRINT CL$"0":OPEN 1,8,0,"*":POKE 781,1:S
    YS 65478:GET A$,A$,A$,A$
B0 340 FOR I=1 TO 7:GET A$,B$,C$,D$:PRINT A$B$C$
    D$:NEXT:PRINT:GET A$,A$,A$,B$

22 345 IF ST THEN SYS 65484:GOTO 355
AE 350 PRINT ASC(A$+W$)+256*ASC(B$+W$):GOTO 340
22 355 CLOSE 1:GOSUB 375:GOTO 120
36 360 OPEN 1,8,15,"10":CLOSE 1:SYS 64738
13 365 DATA 031,008,194,007,158,194,040,052,051,
    041,170,194,040,052,052,041
4E 370 DATA 172,050,053,054,170,052,056,058,086,
    073,082,085,083,000,000,000
E2 375 PRINT:PRINT:PRINT"WCISNIJ RETURN.":POKE 1
    98,0

67 380 GET A$:IF A$<>CHR$(13) GOTO 380
3F 385 RETURN
19 390 F=1:OPEN 1,8,15,"1":INPUT#1,A1$,A2$,A3$,A
    4$:CLOSE 1:IF A1$="00" THEN F=0
60 395 RETURN
    
```

# MAGNETOFON

**Mimo iż stacje dysków są coraz bardziej popularne, magnetofon długo jeszcze będzie służył wielu użytkownikom jako tania pamięć masowa. Warto więc poświęcić mu nieco uwagi i troski aby jego eksploatacja była długa i bezawaryjna.**

Większość użytkowników magnetofonów posługuje się sprzętem fabrycznym. Jest on przeważnie mało zróżnicowany pod względem konstrukcji, a jego podstawowymi elementami są: głowice uniwersalna i kasująca, układ napędowy oraz prosty układ elektroniczny złożony z kilku układów scalonych.

Głowice uniwersalna i kasująca stanowią tu najbardziej wrażliwy i czuły element, są też one najczęstszą przyczyną niedomagań magnetofonu. Zdecydowanie najczęstszym uszkodzeniem objawiającym się radosnym komunikatem LOAD ERROR jest zabrudzenie jednej z głowic. Do szybkiego zanieczyszczenia przyczynia się przede wszystkim stosowanie taśm o marnej jakości, ich niewłaściwe przechowywanie itp. Usunięcie zabrudzenia jest proste i polega na przeczyszczeniu OBU głowic szmatką nasączoną spirytusem (byle nie salicylowym). Należy również obowiązkowo oczyścić z nalotu gumową rolkę napędową i wałek silnika, ponieważ elementy te prowadzą taśmę i jakiegokolwiek osady mogą spowodować przesuwanie taśmy do góry lub w dół, co również uniemożliwia poprawny odczyt programu. Jeżeli zabiegi te nie przyniosą spodziewanych efektów, to w grę może wchodzić niewłaściwe ustawienie głowicy.

Regulacja ustawienia głowicy jest czynnością niezbyt skomplikowaną, wymaga jednak dużej precyzji. Idealne dostrojenie to znaczy takie aby magnetofon odczytywał wszystkie kasetę zapisane na innych magnetofonach nie wchodził w grę, staje się jednak bardziej prawdopodobne gdy głowica jest prawidłowo ustawiona. Samą regulację można wykonać np. za pomocą popularnego programu RECORDER TAPE JUSTAGE Club TAPE JUSTAGE 2). Choć program ten zawiera szczegółową instrukcję obsługi to warto wspomnieć, że śrubą regulacyjną należy kręcić bardzo powoli do momentu uzyskania najbardziej skupionego paska. Na rysunkach poniżej przedstawiam obraz jaki należy uzyskać.

W wypadku gdy nie dysponujemy omawianym programem można spróbować ustawić głowicę „na słuch” lub za pomocą miernika. W pierwszym wypadku konieczne jest posiadanie lub zainstalowanie „podsluchu” i uzyskanie takiego położenia głowicy w którym ton jest najgłośniejszy i zawiera najwięcej dźwięków o wysokich częstotliwościach; w drugim wypadku należy odnaleźć położenie w którym wychylenie wskazówki miernika jest największe. Dobre efekty daje nam połączenie obu metod.

Po operacji ustawiania głowicy, a także niezależnie od niej bardzo pożądane jest okresowe rozmagnesowywanie głowicy. Z braku demagnetyzera posłużyłem się tu zwykłą lutownicą transformatorową w której zamiast grotu zainstalowałem pętlę z o średnicy ok. 20 mm wykonaną z drutu o średnicy 1.2 do 1.5 mm. Głowicę zabezpieczyłem przed bezpośrednim zetknięciem z głowicą za pomocą małej przekładki wykonanej z kartonu. Po włączeniu lutownicy zacząłem ją przybliżać (powoli!) do głowicy z odległości ok. 3 cm a następnie równie wolno ją oddalać. Proces ten powinien trwać kilka sekund i gwarantuje nam całkowite pozabawienie głowicy wszelkich śladów magnetyzmu co oczywiście polepsza eksploatację.

Ostatnim defektem jest jej naturalne zużycie do czego waleń przyczynia się stosowanie nieodpowiednich taśm (np. CrO2 czy METAL), zbyt silny docisk głowicy, zanieczyszczenia i brak konserwacji. Czasami zdarza się, że w uzwojeniu wystąpi zwarcie lub przerwa. Takie wypadki są już znacznie trudniejsze do wychylenia i wymagają wymiany głowicy. Nowa głowica powinna mieć jednak parametry zbliżone jak najbardziej do oryginalnej; pomiar należy przeprowadzić w sposób przedstawiony na rysunku. Z polskich głowic gąbarytowo odpowiadają głowice do magnetofonów MK 122 i MK 125, mają one także zbliżone parametry. Można również pokusić się o zainstalowanie lepszej jakościowo głowicy (np. z utwardzonym czołem) pod warunkiem odpowiednich parametrów.

Niektóre magnetofony mają fabrycznie wbudowane miniaturowe głośniczki umożliwiające odsłuch podczas zapisu i wczytywania programu. Usprawnienie takie jest bardzo proste do wykonania a jego schemat przedstawiony poniżej. W moim magnetofonie zainstalowałem słuchawkę od aparatu dla słabo słyszących; możliwe jest jednak wykorzystanie innych elementów takich jak wkładka mikrofonu piezoelektrycznego, wkładka telefoniczna, słuchawka od magnetofonu typu WALKMAN itp.

**UWAGA: W ŻADNYM WYPADKU NIE WOLNO UŻYĆ DO TEGO CELU ŻADNYCH GŁOŚNIKÓW NISKOOMOWYCH O IMPEDANCJI PONIŻEJ 300-400 OMÓW GDYŻ ZWIĘKSZA TO OBCIĄŻALNOŚĆ UKŁADÓW I MOŻE DOPROWADZIĆ DO USZKODZENIA KOMPUTERA LUB MAGNETOFONU!**

Poniższy rysunek wyjaśnia sposób podłączenia układu. Potencjometr P służy jedynie do ustawienia optymalnego natężenia dźwięku; pamiętaj jednak o tym, że im głośniejszy tym większe jest obciążenie układów elektronicznych.

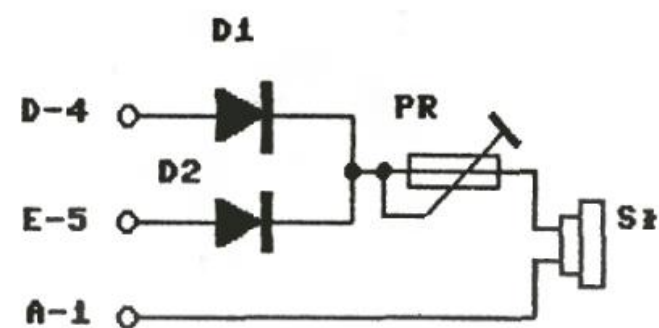


# TEŻ CZŁOWIEK

Na zakończenie kilka porad:

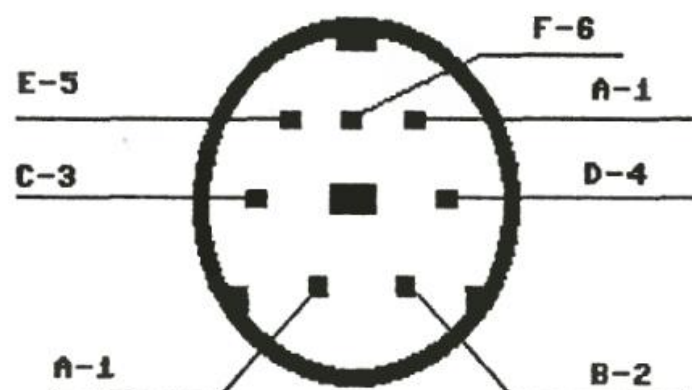
1. Ustawienie komputera, monitora, zasilacza i magnetofonu w „kupie” owocuje zwykle niespodziankami wynikającymi z nakładania się na siebie pól elektromagnetycznych generowanych przez te urządzenia.
2. Kasetę wsuwaj do kieszeni do oporu. Wszelkie luzy powodują zwykle wkręcanie i pętanie taśmy.
3. Gdy z kasety nie korzystasz włóż ją do pudełka w celu ochrony przed kurzem.
4. Gdy taśma się choć raz wkręci, to jej wkręcony fragment nadaje się jedynie do wyrzucenia i należy go odciąć pamiętając jednak, że klejenie rzadko kiedy daje dobre wyniki.
5. Stosuj kasety o maksymalnej długości C-90 typu NORMAL lub FERRO. Kasety innego typu czy dłuższe nie nadają się do Twojego magnetofonu.
6. Ustawianie kaset, w sąsiedztwie zasilaczy, telewizorów, silników elektrycznych i innych urządzeń generujących silne pola elektromagnetyczne kończy się zwykle wyjątkowo nieprzyjemnie.

Zbigniew Kaszycki  
SP 8 IC



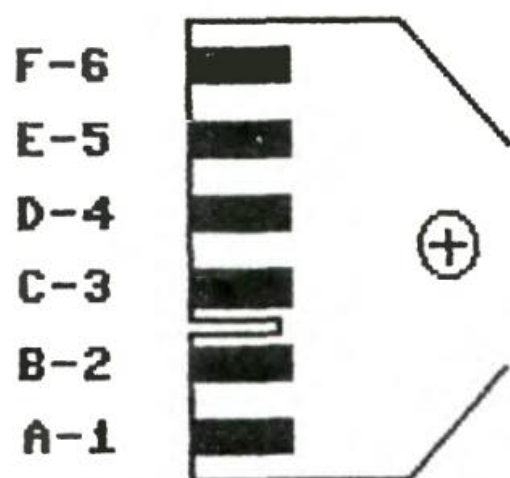
Schemat ideowy podsluchu taśmy

D1, D2 - dowolne diody krzemowe  
PR - potencjometr 500 Ohm  
S1 - słuchawka o impedancji > 400 Ohm



Wtyk magnetofonu 1531

(widok z przodu wtyczki)



Wtyk magnetofonu 1530



Sygnał Turbo — obraz nieprawidłowy



Sygnał Turbo — obraz prawidłowy

# ZNÓW OBRAZEK

Efektowne wykorzystanie rysunku podczas ładowania programu jest często problemem ludzi piszących ambitne programy i chcących nadać im dobrą formę graficzną.

Proste przepisywanie obrazu z pamięci na ekran było już demonstrowane na łamach „Bajtki” nieraz. Należy teraz urozmaicić ten program i nieco zmodyfikować. Na początek spróbujmy pokazać tylko górną, potem tylko dolną połowę rysunku, a na końcu cały.

Wydaje się to trudne do zrealizowania ze względu na skomplikowany sposób wykorzystania pamięci obrazu przez Spectrum. Ekran podzielony jest bowiem na trzy równe części, a na obszarze każdej z nich linie nie są zapalane po kolei. Aby była widoczna tylko połowa rysunku, należy stosować skomplikowane równania.

Wystarczy jednak wpasć na pomysł. Kod maszynowy działa tak szybko, że jest dla oka prawie nieuchwytny. W związku z tym wystarczy przenieść na ekran cały obrazek, a następnie jego dolną część zamazać atrybutami np. na czarno. Potem postąpić dokładnie odwrotnie i wreszcie pokazać rysunek w całości.

Realizuje to pierwszy z podanych programów. Kod maszynowy zajmuje 138 bajtów i dzięki użyciu zmiennych GOTO jest całkowicie relokowany. Należy jednak pamiętać, by po każdej zmianie adresu startowego, wykonany był program w Basic-u, umieszczający porocedurę w pamięci. Początek procedury przechowuje zmienna ADR, zaś adres, od którego umieszczony jest rysunek (np. wczytany z taśmy) — zmienna OBR. Kolor tła przechowuje KOL1, zaś kolor ramki KOL2.

Drugi z podanych programów działa zupełnie inaczej. Rysunek przenoszony jest na ekran tak, jakby układał się on z rozsypanych punktów. W starszych wydaniach programu telewizyjnego „Spectrum” w taki właśnie sposób układały się napisy.

Działanie programu jest dość skomplikowane. Nie przenosi on bowiem rysunku w całości, nie wyświetla też naraz całych bajtów, lecz zapala co ósmy punkt na ekranie, czyli po kolei te same bity w każdym z bajtów. Przy prędkości kodu maszynowego jest to trudne do zaobserwowania, więc efekt jest zadziwiający.

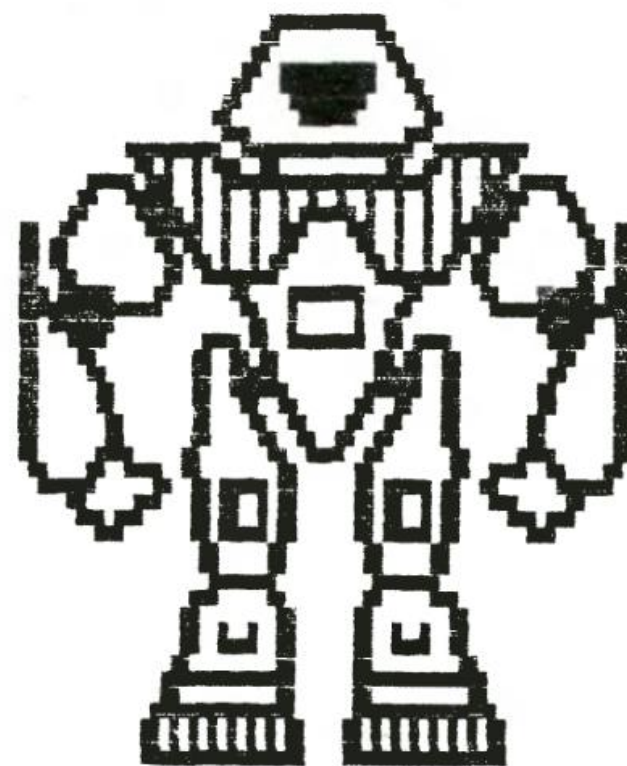
Na końcu rysunek pokrywany jest kolorami, choć można postąpić odwrotnie, najpierw kolorując, a potem rysując treść.

Procedura zajmuje 117 bajtów i jest umieszczana w pamięci w podobny sposób, co poprzednia. Wszystkie zmienne z programu pierwszego spełniają tę samą funkcję w programie drugim.

Te dwa programy są tylko przykładami wykorzystania języka maszynowego do niebywałego urozmaicania rysunków. Pomysłów może być tyle, ilu ludzi zabierze się do ich realizacji. Każde rozwiązanie będzie miało charakter szczegółowy i będzie odrębnym problemem. Tak więc, podając problem obrazków w Spectrum po raz kolejny, mniemam, że czynimy to po raz ostatni.

Michał Sobieszuk

# KLAN SPECTRUM



## PROGRAM 1

```
10 LET KOL1=57: LET KOL2=7: LET ADR=
40000: LET GOTO1=ADR+86: LET GOTO2=ADR
+31: LET OBR=32768: LET OBR=OBR-1: LET
ATR=OBR+6145
20 FOR I=0 TO 116: READ A: POKE (ADR
+I),A: NEXT I
30 DATA 243,205,GOTO1-256*INT(GOTO1/
256),INT(GOTO1/256),62,1,245,205,GOTO2
-256*INT(GOTO2/256),INT(GOTO2/256),241
,7,7,7,254,1,32,244,33,ATR-256*INT(ATR
/256),INT(ATR/256),17,0,88,1,0,3,237,1
76
40 DATA 251,201,33,OBR-256*INT(OBR/2
56),INT(OBR/256),17,255,63,35,19,245,1
23,254,3,32,2,241,201,141,1,255,7,229,
213,245,116,235,182,235,197,6,1,16,254
,193,18,11,120
50 DATA 177,40,11,241,7,7,35,35,19,1
,19,24,228,241,209,225,24,207,62,KOL2
211,254,33,0,64,17,1,64,1,255,23,54,0
237,176,33,0,88,17,1,88,54,KOL1,1,255
,2,237,176,201
```

## PROGRAM 2

```
10 LET KOL1=57: LET KOL2=7: LET ADR=
40000: LET OBR=32768: LET GOTO1=ADR+72
: LET GOTO2=ADR+124: LET GOTO3=ADR+95:
LET GOTO4=ADR+119: LET ATR=OBR+6144:
LET ATR2=OBR+6528
20 FOR I=0 TO 137
30 READ A: POKE (I+ADR),A
40 NEXT I
50 DATA 243,33,0,88,17,1,88,54,KOL1,
1,255,2,237,176,33,OBR-256*INT(OBR/256
),INT(OBR/256),17,0,64,1,0,24,237,176,
33,ATR1-INT(ATR1/256),INT(ATR1/256)
60 DATA 17,0,88,1,128,1,237,176,205,
GOTO1-256*INT(GOTO1/256),INT(GOTO1/256
),205,GOTO2-256*INT(GOTO2/256),INT(GOT
02/256),33,ATR2-256*INT(ATR2/256),INT(
ATR2/256),17,128,89,1,128,1,237,176,20
5,GOTO1-256*INT(GOTO1/256),INT(GOTO1/2
56)
70 DATA 205,GOTO2-256*INT(GOTO2/256)
,INT(GOTO2/256),33,OBR-256*INT(OBR/256
),INT(OBR/256),17,0,64,1,0,27,237,176,
251,201,6,1,197,14,0,6,140,12,197,6,20
5,GOTO3-256*INT(GOTO3/256),INT(GOTO3/2
56)
80 DATA 16,251,193,16,244,193,16,236
,201,197,6,1,197,65,62,KOL2,211,254,16
,252,62,KOL2,211,254,205,GOTO4-256*INT
(GOTO4/256),INT(GOTO4/256)
90 DATA 193,16,238,193,201,201,197,2
13,209,193,201,33,0,88,54,KOL1,17,1,88
,1,255,2,237,176,201
```



# SZYBKI, SZYBSZY, SPECTRUM...

Ten artykuł stanowi uzupełnienie tekstu „Szybki, szybszy, najszybszy”, napisane z myślą o użytkownikach Spectrum. Ponieważ większość rad które są zawarte w tamtym artykule jest ogólna, i niezależna od komputera i języka, radzę przeczytać go najpierw. Już? To możemy zacząć.

Na samym początku chciałbym pokazać, jak badać, które rozwiązania zastosowane w programie są szybsze. Posłużmy się w tym celu prostym programem:

```
10 POKE 236772,0: POKE 23673,0
20 FOR i=1 TO 1000
30 REM miejsce na testowaną instrukcję
40 NEXT i
50 PRINT 256*PEEK 23673+PEEK 23672
```

Pod adresem 23672 i 23673 znajdują się dwa mniej znaczące bajty zmiennej systemowej FRAMES, która jest co 0,02 sekundy zwiększana o jeden. Wykorzystaliśmy to do mierzenia czasu w jakim program się wykonuje — linia dziesiąta to wyzerowanie naszego stopera, linia 50 to odczytanie jego wskazań. Jeśli teraz chcielibyśmy na przykład porównać czas sprawdzania warunków  $a=b$  i  $a \neq b$ , to wstawiamy do programu najpierw linię:

```
30 IF i>500 THEN
(po THEN nie musi być żadnej instrukcji), a potem linię:
```

```
30 IF i>=501 THEN
uruchamiając oczywiście program za każdym razem. Zwróćcie koniecznie uwagę na dwie rzeczy: po pierwsze — warunek jest tak dobrany, żeby był spełniony w połowie przypadków (bo i zmienia się od 1 do 1000), po drugie — warunki są sobie równoważne, to znaczy że zastąpienie w programie jednego drugim nie zmieni algorytmu (oczywiście tylko w przypadku liczb całkowitych). Za pierwszym razem na ekranie pojawia się liczba 369, czyli 7,38 sekundy, za drugim 381, czyli 7,62 sekundy. Wniosek — sprawdza-
```

nie ostrych nierówności jest szybsze. W ten sposób można w każdej chwili sprawdzić którą wersję dowolnej instrukcji jest szybsza. Trzeba tylko koniecznie pamiętać o tym, że samo działanie wykonywane dla różnych danych potrzebuje różną ilość czasu, toteż nie wolno żadnej instrukcji testować dla jednego zestawu danych. (Proponuję sprawdzić szybkość mnożenia  $2*7.59$  i  $1.17*11$ ).

O tym, że nie warto oszczędzać na jednym odejmowaniu, była już mowa w poprzednim artykule, toteż nie będziemy do tego wracać. Powiedzmy sobie za to kilka słów o potęgowaniu. Spectrum wyrażenie  $a \uparrow b$  liczy jako  $EXP(b*LN(a))$ , czyli wykorzystuje w tym celu dwie czasochłonne funkcje. W efekcie jedno podniesienie liczby do potęgi zajmuje tyle samo czasu co 49 (słownie czterdzieści dziewięć!) mnożeń. Dlatego potęgowania najlepiej nie używać, chyba że jest to konieczne.

Następna sprawa, to rozmieszczenie podprogramów. Interpreter trafiając na instrukcję skoku (GO TO lub GO SUB) zaczyna szukać linii o potrzebnym numerze zawsze zaczynając od początku programu. Dlatego też lepiej jest najczęściej używane podprogramy umieścić na samym początku. Czas potrzebny interpreterowi na wyszukanie określonej linii jest zależny tylko od tego, ile linii znajduje się przed poszukiwaną, nie ma natomiast znaczenia jakiej są one długości. Ponadto wykonanie dwóch instrukcji znajdujących się w jednej linii jest nieco szybsze niż wykonanie tych samych dwóch instrukcji umieszczonych w kolejnych liniach. Wniosek nasuwa się sam — tam gdzie jest to możliwe warto umieszczać po kilka instrukcji w jednej linii (oczywiście trzeba umieć zachować umiar, za długa linia jest niewygodna w edycji i nieczytelna).

Podobnie należy postępować ze zmiennymi — są one zapamiętane w takiej kolejności, w jakiej były podstawiane, i za każdym razem szukane od początku, toteż dobrze jest zadeklarować je w jednej z pierwszych linii programu w odpowiedniej kolejności — jako pierwsze te zmienne, które będą

używane najczęściej — np.  $LET a=0$ :  $LET b=0$  itd. Również długość nazwy ma swoje znaczenie — korzystniej jest używać krótkich nazw niż długich.

Teraz kilka słów na temat instrukcji warunkowych. Po pierwsze, każdą zmienną można potraktować jako zmienną logiczną (a w każdym razie interpreter na to pozwala), mająca wartość TRUE gdy jest różna od zera, i wartość FALSE gdy jest równa zero. Operacje logiczne są szybsze niż arytmetyczne, toteż od instrukcji  $IF a=0 THEN$  szybsza jest  $IF a THEN$ , a od instrukcji  $IF a=0 THEN$  szybsza jest  $IF NOT a THEN$ . Istniejąca na Spectrum możliwość zastąpienia instrukcji:

```
IF a>0 THEN LET a=a+p
```

instrukcją:

```
LET a=a+(p AND a>0)
```

jest z punktu widzenia szybkości działania niekorzystna.

O sprawdzaniu warunków arytmetycznych była już mowa — teraz tylko krótkie rozszerzenie. Porównywanie  $a=b$  i  $a \neq b$  (lub  $a=b$ ) są mniej więcej tak samo szybkie, wolniejsze od nich jest  $a \neq b$ , a najwolniejsze  $a=b$  (lub  $a \neq b$ ).

Ostatnią instrukcją na którą zwróćmy uwagę jest PRINT. Najszybciej drukowane są zmienne łańcuchowe, przy których interpreter ma za zadanie tylko wywołać odpowiednią procedurę. Wszystkie rozwinięcia instrukcji PRINT takie jak PRINT AT, INK czy PAPER, wymagają znacznie więcej czasu. Dlatego też czasem przydatnym może się okazać wcześniejsze przygotowanie sobie odpowiedniego łańcucha do wydrukowania, korzystając przy tym z kodów kontrolnych — CHR\$ 22 zamiast AT, CHR\$ 16 zamiast INK itd. (Szczegóły są opisane w instrukcji do Spectrum).

Na tym koniec. Oczywiście nie zajęliśmy się wszystkimi instrukcjami, ale byłoby to pozbawione sensu. Nie ma jednej recepty na szybkie programy, bo prawie zawsze można znaleźć jakąś wyjątkową sytuację, w której nie sprawdzą się znane chwytły. Dlatego zachęcam do szukania własnych rozwiązań, chociażby posługując się zaproponowanym na początku programem jako narzędziem.

Marcin Borkowski

## MOŻLIWOŚCI JESZCZE WIĘKSZE

Poprzednio opisywana seria przystawek do ZX Spectrum oferowana jest przez firmę Datel Electronics. Tu muszę zaznaczyć, iż ani poprzedni, ani ten artykuł nie są reklamą firmy, lecz informacjami i ciekawostkami na temat rozwoju sprzętu w Wielkiej Brytanii. Tak więc proszę nie zasypywać nas listami z błaganiami o adres firmy, gdyż prośby te spełnione nie będą.

Tym razem zajmujemy się wachlarzem przystawek oferowanych przez firmę Romantic Robot. Najważniejszą ofertą zdaje

się być zestaw VIDEOFACE. Za jedyne 44 funty można otrzymać przystawkę, dzięki której możliwe jest przesłanie dowolnego obrazu z kamery video lub magnetowidu do pamięci Spectrum. Format przesłanego obrazu odpowiada oczywiście standardowi — 16384 bajty w rozdzielczości  $256*192$  i ośmiu kolorach. Rysunki mogą zostać wydrukowane, nagrane na taśmę lub inny nośnik, zmodyfikowane za pomocą programu graficznego (np. ART STUDIO), animowane (VIDEOFACE pamięta naraz 6 screenów). Dalsze zalety to duża szybkość transmisji (3 obrazy na sekundę) i niezwykle łatwa obsługa zestawu. Całość wydaje się być ciekawą i unikalną propozycją.

30 funtów trzeba zapłacić za przystawkę o nazwie MULTIPRINT. Pasuje ona do wszystkich modeli Spectrum, współpracuje z Interface 1 i innymi interface'ami, posiada 8KB ROM i tyleż RAM. Za pomocą umieszczonego na niej przycisku można „zamrozić” w dowolnym momencie wykonywanie programu i wybrać jedną z możliwości — nagranie obrazka, wydruk obrazka, wydruk programu lub szerokie menu opcji użytkowych. MULTIPRINT daje się również doskonale wykorzystywać jako interface standardu Centronics.

Powstała cała seria przystawek typu MULTIFACE. Dotychczas znane wersje to MULTIFACE 1 i MULTIFACE 128, zaś jednym z ostatnich hitów jest MULTIFACE 3. Pierwsza z nich współpracuje ze Spectrum 48 i +, wersja druga ze Spectrum 128 i Spectrum +2, zaś trzecia tylko ze Spec-

trum +3. Każda z przystawek kosztuje ok. 40 — 50 funtów i oferuje bogaty zestaw możliwości.

Przed wszystkim możliwe jest zatrzymanie dowolnego programu w dowolnym miejscu i nagranie go na nośnik, przejrzanie pamięci, zmiana treści (np. wpisanie POKE-ów na nieśmiertelność), nagranie lub wydruk treści ekranu, wreszcie kontynuacja przerwanego programu. Dodatkowo MULTIFACE 3 posiada zestaw procedur narzędziowych do współpracy z napędem dyskietek. Dzięki niemu można m.in. kopiować gry z taśmy na dyskietkę, z taśmy na taśmę w turbo i korzystać z wielu innych dogodnych usprawnień.

Angielski miesięcznik „CRASH” reklamując MULTIFACE 3 orzekł „To dobra przyczyna, by kupić Spectrum +3”.

Ostatnią propozycją firmy Romantic Robot jest przystawka LIFEGUARD, oferowana za 7 funtów. Jest to wspaniałe rozwiązanie dla graczy, którzy lubią „być nieśmiertelnymi” w ulubionych grach. LIFEGUARD musi być podłączony do Spectrum za pomocą którejś z przystawek MULTIFACE, co automatycznie zwiększa koszt tej zabawy. Naciśnięcie umieszczonego na MULTIFACE przycisku powoduje uaktywnienie się przystawki LIFEGUARD. Może ona znaleźć i zainstalować nieśmiertelność, nieograniczoną ilość amunicji itp. Gracze będą zachwyceni możliwościami, lecz na pewno nie ceną.

Marcin Przasnyski

INTERFACE

DO PCW 8256/8512

FIRMY SCA SYSTEMS LTD



**Zamawianie sprzętu komputerowego w zagranicznych firmach wysyłkowych ma swoje wady i zalety. Jest to sposób niekrępujący, ponieważ płacimy i wymagamy. Niestety, czasem musimy poczekać nawet parę miesięcy nim otrzymamy zamówiony gadżet.**

Z podobnym problemem zetknąłem się przy zamawianiu interface'u CPS 8256 do Amstrada w jednej z angielskich firm wysyłkowych. Okazało się, że już chyba nie jest on produkowany, albo firma nie potrafiła go nabyć, bo do tej pory nie otrzymaliśmy go. Tłumaczono nam, że w Wielkiej Brytanii, podobnie jak i u nas zdarzają się niedobory pewnych towarów i że w związku z tym musimy poczekać.



Ponieważ miałem pewne opory przed przyjęciem tej linii rozumowania, zwróciłem się do swojego kolegi, zamieszkałego w Londynie, o nabytanie takiego lub podobnego interfejsu. W ciągu 3 tygodni otrzymałem przesyłkę zawierającą potrzebne mi urządzenie. Po zapoznaniu się z dokumentacją stwierdziłem, że przysłany interfejs firmy SCA Systems Ltd jest znacznie lepszy od CPS 8256, o którym poprzednio pisałem w Bajtku [1].

Przejdźmy do opisu samego sprzętu. Produkt firmy SCA w białej, solidnie wyglądającej obudowie, rozmiarami odpowiada interfejsowi CPS 8256 [1] i w ten sam sposób, jak on, montowany jest do komputera. Wewnątrz oprócz złącza RS 232C i Centronics znajduje się zegar czasu rzeczywistego, podtrzymywany bateryjnie. Gniazdo interfejsu RS 232 typu Canon 25 zawiera również sygnały drugiego, uproszczonego RS'a. Do podłączenia drukarki służy gniazdo typu Amphenol. Oprócz sprzętu dostajemy krótką instrukcję obsługi oraz niezbędne oprogramowanie.

## DYSKIETKA FIRMOWA

Na dostarczonej, razem z interfejsem, dyskietce 3 calowej znajduje się siedem zbiorów:

TIME.COM TIME.SUB GETIME.BAS CLOCK.COM  
TERM.SUB TERMX.SUB RSTEST.BAS

Najważniejszy z nich — TIME.COM — zapewnia dostęp do podtrzymywanego bateryjnie zegara czasu rzeczywistego. Wywołanie programu TIME zastępuje ręczne ustawienie wewnętrznego zegara, wykonywane zwykle przy pomocy systemowego programu DATE.COM. Natomiast komenda TIME z opcją s umożliwia zmianę daty i czasu zegara baterijnego.

Zbiór TIME.SUB zawiera dwa polecenia: TIME i SETSIO. Dodanie tych komend do zbioru PROFILE.SUB spowoduje automatyczne ustawienie właściwego czasu zegara systemowego i zada standardowe parametry transmisji złącza RS 232C. Na startowej dyskietce systemowej muszą znajdować się dwa zbiory z dyskietek firmowych CP/M'u: SUBMIT.COM i SETSIO.COM. Krótki program zawarty w GETIME.BAS pozwala na dostęp do zegara systemowego w Basic'u Mallarda. Program CLOCK.COM demonstruje na ekranie komputera chodzący zegar analogowy. Poprawne działanie tego programu wymaga skopiowania na dyskietkę firmową zbiorów GSX'a: ASSIGN.SYS, GSX.SYS, DDSCRE-EN.PRL.

Wsadowy zbiór TERM.SUB zawiera komendy:

DEVICE CONOUT := SIO  
DEVICE CONIN := SIO  
DEVICE LST := CEN

i pozwala na korzystanie z Amstrada przy pomocy dodatkowego terminala z drukarką typu Centronics. Powrót do pierwotnej klawiatury i ekranu zapewnia zbiór TERMX.SUB, zawierający dwa polecenia:

DEVICE CONOUT := CRT  
DEVICE CONIN := CRT

Do uruchomienia tych zbiorów wsadowych konieczne są programy SUBMIT.COM i DEVICE.COM.

W pliku RSTEST.BAS znajduje się krótki program w Basic'u umożliwiający dostęp do drugiego złącza RS. W systemie CP/M nie przewidziano takiego rozwiązania i obsługa dodatkowego RS'a jest pominięta w programach systemowych.

## ZŁĄCZE CENTRONICS

W interfejsie firmy SCA zastosowano uproszczoną wersję standardowego złącza równoległego typu Centronics. Jego opis znajduje się na rys. 1. Złącze to pozwala korzystać na komputerze Amstrad PCW z typowych drukarek, jakie występują u nas na rynku: SG 10, SG 15, NX 10, NX 15, LC 10 itp. Możliwe jest również podłączenie dowolnego plotera wyposażonego w Centronics, np. Roland DXY 880, Sony C41 [2] itp. Programowy dostęp do złącza równoległego zapewnia komenda systemowa DEVICE.COM. W

CP/M'ie mamy do czynienia z urządzeniami logicznymi i fizycznymi. Program DEVICE służy do wzajemnego konfigurowania tych urządzeń. Standardowo urządzenie logiczne LST ma przypisaną urządzenie fizyczne LPT, tzn. drukarkę PCW. Napisanie DEVICE LST:=CEN powoduje zmianę tego przypisania. Wydruki zostają skierowane na drukarkę podłączoną do Centronics'a. Powrót do drukarki PCW następuje po komendzie DEVICE LST:=LPT.

## ZŁĄCZE RS 232C

Interfejs firmy SCA zawiera dwa złącza szeregowo typu RS-232C. Opis sygnałów umieszczono na rys. 2. Przy pomocy złącza szeregowego możliwe jest dołączenie wielu różnych urządzeń zewnętrznych do Amstrada PCW:

1. drugi komputer
2. modem
3. drukarki, plotery wyposażone w złącze RS-232C
4. mysz
5. programator EPROM'ów

Szczególnie interesujące w naszych warunkach jest podłączenie Amstrada do komputerów typu IBM PC. Pozwala to na transfer zbiorów między tymi dwoma maszynami [3], a także na pracę jednego z nich jako terminal drugiego. Typowy kabel jaki należy w tym przypadku zastosować przedstawiono na rys. 3. Do emulacji terminala VT52 na PCW służy program MAIL232.COM. Program ten pozwala także na transfer zbiorów między dwoma Amstradami. Do współpracy z innymi urządzeniami zewnętrznymi służą następujące programy systemowe: DEVICE.COM, SETSIO.COM, PIP.COM.

Komenda SETSIO ustawia parametry transmisji i ma składnię:

SETSIO T<sub>n1</sub> R<sub>n2</sub> BITS<sub>n3</sub> STOP<sub>n4</sub> PARITY<sub>p1</sub>  
XON<sub>p2</sub> HANDSHAKE<sub>p3</sub> INTERRUPT<sub>p4</sub>  
gdzie  
n1, n2 szybkość nadawania i odbioru w bodach,  
n3 liczba bitów w słowie,  
n4 liczba bitów stopu  
p1 parzystość (EVEN, ODD, NONE)  
p2 programowy 'handshake' (ON/OFF)  
p3 sprzętowy 'handshake' (ON/OFF)  
p4 przerwania (ON/OFF).

Komenda DEVICE.COM, podobnie jak i w przypadku złącza Centronics, stosowana jest do zmiany wzajemnych przypisań urządzeń logicznych i fizycznych. Korzystanie z drukarki lub plotera podłączonego do RS'a wymaga napisania: DEVICE LST:=SIO. Powrót do drukarki PCW po komendzie DEVICE LST:=LPT.

Transfer zbiorów między komputerami zapewnia komenda PIP.COM:

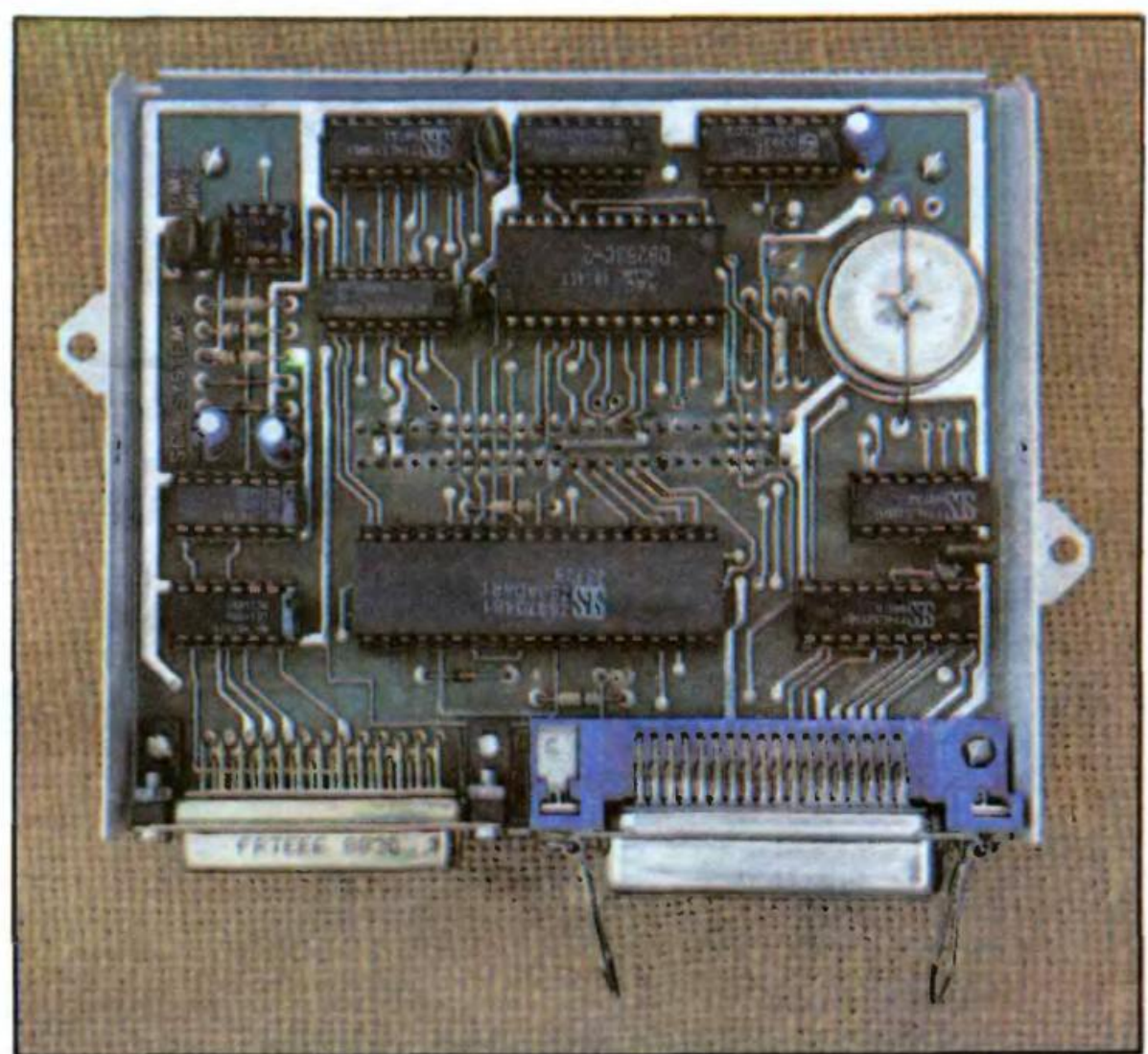
1. PIP AUXOUT:=nazwa\_zbioru, przesyła zbiór z PCW na inny komputer, a
2. PIP nazwa\_zbioru=AUXIN:, odbiera zbiór transmitowany przez inny komputer do PCW.

Jeśli chcemy do Amstrada podłączyć jakiś inny komputer w charakterze terminala, np. ZX Spectrum, to musimy posłużyć się serią poleceń zawartą w zbiorze wsadowym TERM.SUB. Powrót do pierwotnych urządzeń fizycznych (ekran i klawiatura) zapewnia wykonanie zbioru wsadowego TERMX.SUB.

Do Amstrada PCW można podłączyć typową dla komputerów IBM PC mysz, np. typu Witty C-400. Niestety nie widziałem jeszcze oprogramowania do zapewnienia takiej współpracy.

## ZEGAR PODTRZYMANY BATERYJNIE

O ile złącza RS 232C i Centronics są prawdziwym „oknem na świat” dla PCW, to baterijny zegar zapewnia inny komfort pracy. Po włączeniu komputera do sieci konieczne jest zadanie właściwego czasu na wewnętrznym zegarze systemowym. Bez interfejsu firmy SCA trzeba było robić to ręcznie przy pomocy programu DATE.COM z opcją SET. Nie jest to specjalnie wygodne. Z kolei aktualna data może być zapisywana w katalogu



dyskietki przy tworzeniu danego zbioru. Zapewnia to większy porządek w pracy prawdę trudno zapamiętać, kiedy ostatni raz zmienialiśmy dany zbiór. System operacyjny CP/M, dzięki programom INITDIR.COM i SET.COM pozwala wprowadzić datowanie zbiorów [4]. Baterijny zegar przepisujący właściwy czas do zegara wewnętrznego ułatwia istotnie pracę z komputerem. Wywołanie TIME s, umożliwia zmianę czasu w zegarze zewnętrznym. Do zegara systemowego mamy dostęp przez program DATE.COM spod systemu, a przy korzystaniu z Basic'a możemy posłużyć się przykładowym programem GETIME.BAS. Jeśli chcemy to zrobić w programach pascalcowych, to w „Bajtku” 1,89 [5] znajdziemy też odpowiedni opis.

## PODSUMOWANIE

Testowany interfejs firmy SCA jest doskonałym uzupełnieniem komputera Amstrad PCW 8256/8512 i może być polecony każdemu użytkownikowi tego sprzętu. Ze względu na podtrzymywany bateryjnie zegar, dodatkowe, drugie złącze RS-232C i porównywalną cenę (ca. 60 funtów) jest on zdecydowanie lepszy od, oferowanego przez firmę Amstrad, interfejsu CPS 8256.

Oprócz opisywanego interfejsu można w komputerze PCW 8256 rozszerzyć pamięć RAM do 512KB i dodać napęd 5 i 1/4 cala (1.2MB) oraz dysk twardy 20-30MB. To ostatnie rozwiązanie jest oferowane przez jedną z polskich firm. Niestety redakcji mimo wielu prób nie udało

na komputerze. Po paru miesiącach nasię nakłonić wspomnianej firmy do zaprezentowania go na łamach „Bajtku”.

Jonasz Mayer

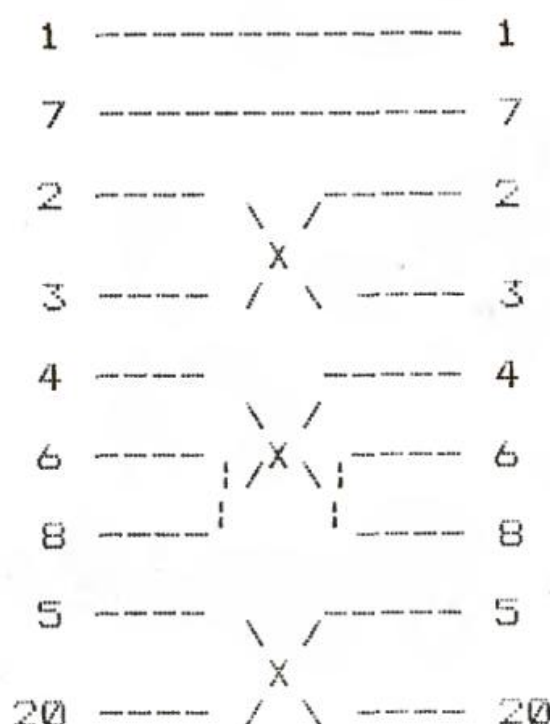
Testowany interfejs otrzymałem z Anglii dzięki uprzejmości pana Artura Janowskiego zamieszkałego w Londynie. Dziękuję.

## LITERATURA.

- [1] JM., „CPS 8256...”, Bajtek 6,88
- [2] J. Mayer, „Test plotera PRN C41 firmy SONY”, Bajtek 3,89
- [3] J. Mayer, „COMHEX...”, Bajtek 6,88
- [4] J. Młodzki, „Sztuczki i chwyt”, Komputer 4,88
- [5] J. Mayer, „Czas i data w Turbo Pascalu”, Bajtek 1,89

Nr przewodu	Nazwa sygnału
2	TX (TRANSMIT DATA)
3	RX (RECEIVE DATA)
4	RTS (READY TO SEND)
5	CTS (CLEAR TO SEND)
7	GND (GROUND)
8	DCD (DATA CARRIER DETECT)
14	TX(2) (TRANSMIT DATA(2))
16	RX(2) (RECEIVE DATA(2))
20	DTR (DATA TERMINAL READY)
22	RI (RING INDICATOR)

Rys. 2. Opis złącza RS-232 C

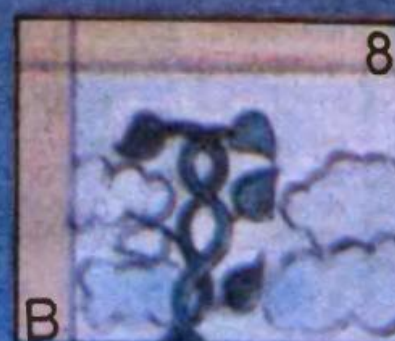
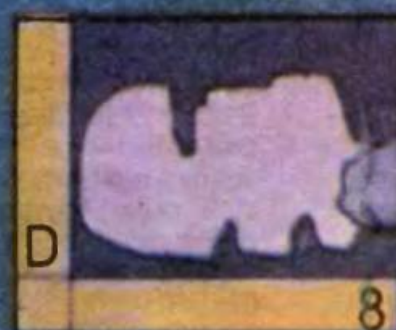
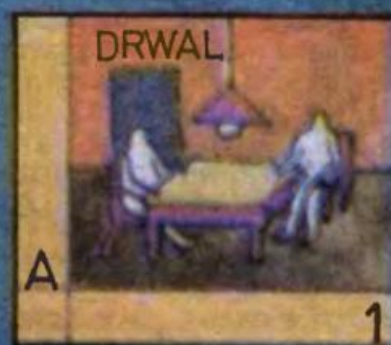
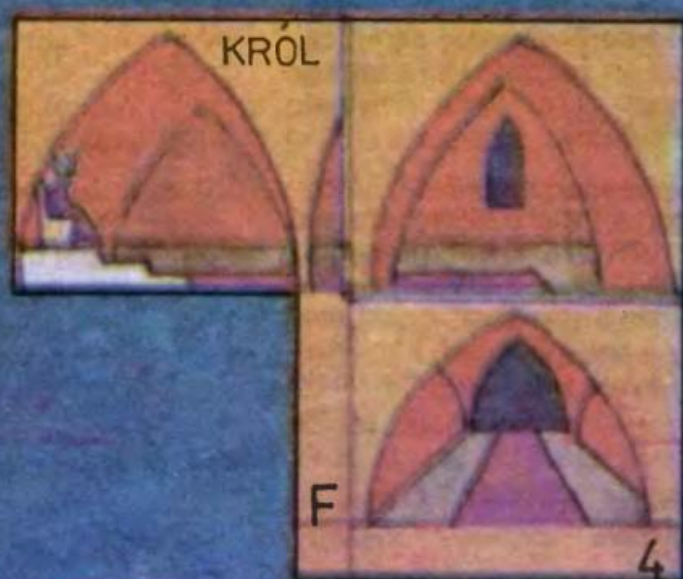
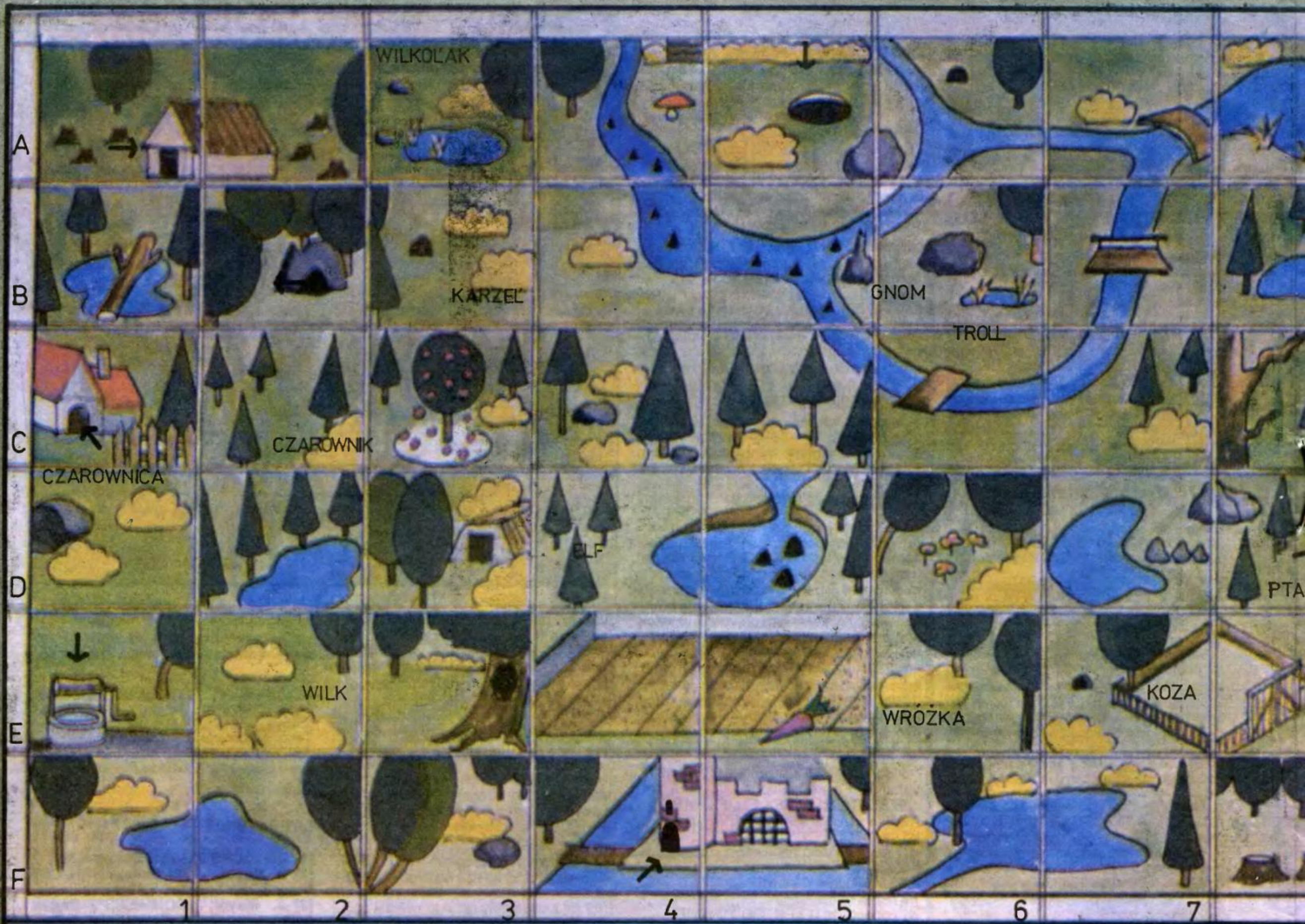


Nr przewodu	Nazwa sygnału	NR przewodu	Nazwa sygnału
1	STB'	19	GND
2	DATA 1	20	GND
3	DATA 2	21	GND
4	DATA 3	22	GND
5	DATA 4	23	GND
6	DATA 5	24	GND
7	DATA 6	25	GND
8	DATA 7	26	GND
9	DATA 8	27	GND
10	NC	28	GND
11	BUSY	29	NC
12	PAPER OUT	30	NC
13	NC	31	NC
14	GND	32	GND
15	NC	33	NC
16	GND	34	NC
17	NC	35	NC
18	NC	36	NC

Rys. 1 Opis złącza Centronics.  
GND — masa sygnałowa,  
NC — niepodłączone, apostrof — negacja sygnału.

Rys. 3. Opis typowego kabla do transferu zbiorów złączem RS-232C





**Bojtek**



# KING'S QUEST

Ta gra nie ma legendy; ona sama jest legendą. Legendą o chylącym się ku upadkowi królestwie i o młodym Sir Grahamie, który ma przywrócić mu świetność. Na tronie zasiada król Edward Zycziwy, bardzo stary i schorowany. Jest tak słaby, że nie może nawet podnieść się ze swego złotego tronu. Nie może więc włączyć królestwem, które pozostawione sobie, coraz bardziej upada. Król wybrał więc młodego Grahama, jako godnego następcę tronu. Aby udowodnić, że jest godzien swej przyszłej roli, musi on zdobyć trzy magiczne przedmioty — lustro przepowiadające przyszłość, tarczę chroniącą przed każdą bronią i skrzynię, zawsze pełną pieniędzy. Do pomocy, Sir Graham ma jedynie własny spryt.

Tym, którzy chcą ukończyć tę grę od razu, przyda się ta ściągawka:

1. Przesuń kamień (MOVE ROCK) w F3 i wydobyj z dziury sztylet (TAKE DAGGER).
2. W E1 wskocz do studni (ENTER WELL), pływaj (SWIM) i zanurkuj (DIVE). Wpływ w dziurę z lewej strony, podejdź do smoka i rzuć w niego sztyletem (THROW DAGGER).
3. Zabierz lustro (MIRROR) i wróć na powierzchnię, wspinając się po linie (CLIMB ROPE).
4. W E3 wejdź na dąb (CLIMB TREE), zabierz z gniazda jako (EGG) i ostrożnie zejź.
5. W D6 zerwij czterolistną koniczynę (TAKE CLOVER) i pójź do D8.
6. Stań na środku i w odpowiedniej chwili podskocz (JUMP), by złapać się ptaka.
7. Spadniesz na wyspę. Z A4 weź grzyb (MUSHROOM) i wskocz do dziury w A5.
8. Szczurowi daj jajko (GIVE EGG) i otwórz drzwi. Pójź w dół, obok tronu stoi tarcza (SHIELD), którą powinienes zabrać.
9. Pójź w lewo do dziury, zjedz grzyb (EAT MUSHROOM) i przejdź przez dziurę. Znajdziesz się w B2.
10. Spod drzewa w C3 zabierz szyszkę (WALNUT), którą otwórz (OPEN WALNUT). Otwartą daj trollowi na moście w C6.
11. Odezwij się do gнома (SPEAK GNOME) w B6. On każe, byś zgadł jego imię, odpowiedź „IFNKOVHGROGHPRM”, dostaniesz fasolę (BEANS).
12. Zasadź fasolę w bagienku w B8 — urosnie wielkie drzewo fasolowe. Wejdź na szczyt i pójź w prawo.
13. Stań za drzewem tak, żeby olbrzym nie widział Cię i poczekaj, aż zaśnie. Wtedy podejdź do niego i zabierz mu skrzynię (CHEST).
14. Wróć do zamku, pójź do króla i stojąc przed tronem ukłoń się (BOW KING). Król wstanie z tronu, by Cię powitać, lecz upadnie i umrze. Podejdź do tronu — teraz Ty jesteś królem!

Największą przyjemność dają jednak samodzielne rozwiązywanie gry.

W przypadku King's Quest jest to prawie niemożliwe. Ci, którzy będą chcieli próbować, muszą poświęcić na to minimum tydzień. Pożądana jest znajomość angielskiego, ale przyda się też słownik. Dobrze będzie także przeczytać poniższe wskazówki oraz zerknąć na tabelkę rozmieszczenia przedmiotów.

— Miska, leżąca w C4 napienia się sama gulaszem, po wypowiedzeniu zaklęcia FILL. Drwal za pełną miskę odda skrzypce, na których można grać.

— Szczur nie pogardzi też złotą szyszką, a troll na moście przyjmie również torbę pełną diamentów.

— Jaskinia w D8 jest zawałona głazem. Odwała go smok uciekając przed wodą, którą można go połączyć.

— Zamiast czekać, aż olbrzym zaśnie, można zabić go z procy naładowanej kamykami.

— Król krasnoludków ma berło i tarczę, jednak by wejść do jego królestwa, trzeba mieć czterolistną koniczynę.

— Czarownica ma w szafce pyszny ser (szwajcarski!).

— Gdy nie zgadnie się imienia gнома, daje on klucz, którym otworzyć można tunel w D3, prowadzący w chmury.

— Po lesie biegać różne stworzenia i osobniki: wilkołak (zabija), karzeł (kradnie zdobycz), czarownik (zamraża), czarownica (wiezi w chatce), elf (daje pierścień niewidzialności), wroźka (jej czar chroni przed niebezpieczeństwami), troll (pilnuje mostu), gnom (waleś się wokół legowiska), ptak (przenosi na wyspę) i koza (pasie się w zagrodzie). Dodatkowo w jaskini urzęduje smok, w chmurach olbrzym, w chacie drwal z żoną, w podziemiach szczur i krasnoludki. Zwierzęta przebywające w lesie nie są groźne, gdy za Grahamem idzie koza. Należy zwabić ją marchewką, wyrwaną z pola.

— Jeśli Graham wpadnie do wody, tonie po pięciu sekundach, jeśli nie popłynie. Pod wodą może przebywać tylko dwadzieścia sekund.

— Niektóre funkcje mogą być wydawane automatycznie, np. 3 — nagranie aktualnej pozycji, 5 — odtworzenie nagranej pozycji, 7 — rozpoczęcie gry od początku, 9 — powtórzenie ostatniego rozkazu, 0 — podskok, + — pływanie, — — kłęknięcie.

— Po wejściu do pomieszczenia, najpierw rozejrzyj się (LOOK AROUND), przydatne jest też oglądanie przedmiotów.

— Nie używaj słów „do”, „dla”, „od”, „z”, „na” itp. Są one pomijane. Na przykład rozkaz „SPEAK KING” powoduje to samo, co „SPEAK TO KING”.

co?	gdzie?	po co?
BEANS (fasola)	B6	zasadzić w B8
BOUL (miska)	C4	dać drwalowi w A1
BUCKET (wiadro)	E1	napętnić wodą
CARROT (marchew)	E5	zjeść lub dać kozie
CHEESE (ser)	C1	zjeść
CHEST (skrzynia)	w chmurach (B8)	dla króla
CLOVER (koniczyna)	D6	chroni przed krasnoludkami
DAGGER (sztylet)	F3	odciąć wiadro, zabić smoka
EGG (jajo)	E3	dać szczurowi
FIDDLE (skrzypce)	A1	grać
KEY (klucz)	B6	otworzyć drzwi w D3
MIRROR (lustro)	w studni (E1)	dla króla
MUSHROOM (grzyb)	A4	zmniejsza
NOTE (zapiski)	C1	czytać
PEBBLES (kamyki)	B4	strzelać z procy
POUCH (torba)	F8	dać trollowi
RING (pierścień)	D4	czyni niewidzialnym
SCEPTRE (berło)	w podziemiach (A5)	to Twoje!
SHIELD (tarcza)	w podziemiach (A5)	dla króla
SLING (proca)	w chmurach (B8)	zabić olbrzyma
WALNUT (szyszka)	C3	dać trollowi
WATER (woda)	dowolne jezioro	pić lub oblać smoka

Firma: Sierra On-Line  
Komputer: ZX Spectrum 48/+, Commodore 64/128, IBM PC

GEN & LUKE





# 10

## BAJTKOWA LISTA PRZEBOJÓW (3/89)

Dziękuję za nadsyłane do redakcji opisy gier i mapy. Większość z nich jest całkiem udana i z pewnością będą wykorzystane. Mam tylko prośbę: nie nadsyłajcie kserokopii opisów krążących na giełdach, ani też innych plagiatów. Otrzymałem bowiem już niejednego list, w którym znajdowały się opisy przepisane żywcem z „Bajtki”.

Na marcowe notowanie otrzymaliśmy 2507 propozycji na 115 tytułów gier.

	ATARI	AMSTRAD	COMMODORE	SPECTRUM
1 GRYZOR		x	x	x
2 SKATE CRAZY		x	x	x
3 PLATOON	x	x	x	x
4 MOUSE TRAP	x		x	
5 GREEN BERET	!	x	x	x
6 UNIVERSAL HERO	↑	x	x	x
7 NOSFERATU	↓		x	
8 STRONG MAN	!	x	x	x
9 STRIKE FORCE COBRA		x	x	x
10 ARNY MOVES	!	x		x



# ZŁOTA DZIESIĄTKA ROKU 1988

Tak jak przed rokiem, opracowaliśmy zestawienie najpopularniejszych gier, na podstawie Bajtkowej Listy Przebojów. Za pierwsze miesiące na Liście gra otrzymała 10, zaś za ostatnie 1 punkt. Oto, co się okazało:

## 1 SABOTEUR II Durell Software

Chociaż pomysł wzorowany jest na mniej popularnej pierwszej części gry, to jednak okazał się on przebojem. Prawdopodobnie o sukcesie zadecydowała wielka różnorodność gry, możliwość wyboru rodzaju misji a już napewno doskonała grafika i wspaniałe opracowanie. Mimo, że nie powstanie już SABOTEUR III, to czekamy na kolejny hit w wykonaniu firmy Durell i jej czołowego programisty — Mike'a Richardsona.



## 2 SOLO FLIGHT Micro Prose

Tak, jak każdy produkt firmy Micro Prose, tak i ten jest rewelacją w dziedzinie gier symulacyjnych. Co do innych wytwórców tej firmy, to wystarczy powiedzieć, że są hitami. Wymienić tu należy Silent Service, F-15 Strike Eagle, Mig Alley Ace, Airborne Ranger, Kennedy Approach, Gu Ship oraz ostatni przebój — Night Raider.

SOLO FLIGHT jest specyficzną symulacją lotu małym samolotem pocztowym nad terytorium Stanów Zjednoczonych. Wielka różnorodność tras i sporo opcji gry dają duże możliwości przyszłym pilotom.



## 3 ARKANOID Imagine Software

Pierwszą grą, z którą spotykają się użytkownicy Spectrum, jest WALL. Krótki, napisany w Basic-u program pozwala na wielogodzinną zabawę w odbijanie piłki od ściany. Pierwowzór został tak rozbudowany, że stał się bardziej skomplikowany od niejednej strzelaniny. Droga do Arkanoid była długa. Najpierw Krakout, potem Krakout II, następnie Thrust i dopiero potem Arkanoid. Powstała i wersja o tytule Arkanoid II — Batty, lecz nie stała się aż tak popularna. Dzięki swym wersjom na wszystkie niemal komputery, Arkanoid jest grą, w którą grają wszyscy.

## 4 SKYFOX

Ma ponad trzy lata, co dla gry jest wiekiem niemalże śmiertelnym. Mimo to wciąż trzyma się wysoko. Jest ona połączeniem symulacji ze strzelaniną. Do dyspozycji gracza pozostaje nowoczesny myśliwiec niszczący wszelkie możliwe cele. Właściwie i umiejętnie użyty staje się groźnym narzędziem, zaś SKYFOX ma pomóc w jego opanowaniu.

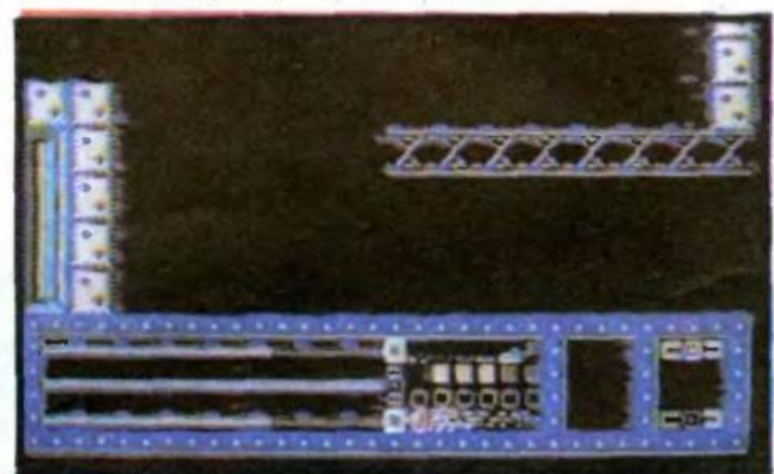
## 5 THE WAY OF THE EXPLODING FIST III Melbourne Hous

Mogłoby się zdawać, że produkt z liczbą trzy w nazwie jest kopią pierwowzoru nie niosącą niczego nowego. Tak jednak nie jest. Producenci pierwszej części Drogi Eksplodującej Pięści nie dublowali pomysłu. Druga część gry była samouczącym się rozwinięciem pierwszej, zaś trzecia jest

połączeniem dwóch poprzednich z typową labiryntową wędrownką, w której do wykonania jest określone zadanie. Wszystko to czyni EXPLODING FIST III popularną i lubianą rozrywką, która wciąż trzyma się wysoko.

## 6 EQUINOX Mikro Gen

Jak zawsze, tak i tu Mikro Gen gwarantuje doskonałą zabawę, oprawioną w wyszukane ramy grafiki i muzyki. Tym razem gracz pilotuje małego robota o kulistym kształcie, który ma za zadanie oczyścić kopalnię planety Equinox z promieniotwórczych ładunków. Warto zauważyć, że nawet w wersji na Spectrum słychać muzykę Jeana Michela Jarre, z płyty „Equinox”.



## 7 THE SECRET DIARY OF ADRIAN MOLE Level 9

To była pierwsza rewelacja. Dziennik czternastoletniego Anglika Adriana Mole'a przekazany w postaci gry komputerowej. Tekst ilustrowany jest wieloma rysunkami, co jeszcze bardziej uatrakcyjniło grę. Pożądana jest tu znajomość języka angielskiego, lecz nie musi być ona perfekcyjna. Doprowadzenie gry do końca na pewno wzbogaci słownictwo grającego i przyczyni się do poprawy jego stylizacji w tym języku.

## 8 SPLITTING IMAGES Domark

Zaczerpnięty z programu telewizji angielskiej pomysł zaowocował naprawdę dobrą grą, połączeniem wyszukanej grafiki i starannej oprawy dźwiękowej oraz niegłupiego tematu. Zadaniem gracza jest ułożenie z 20 kawałków wizerunku jednego ze sławnych ludzi. Łamigłówkę uatrakcyjnią przeszkody w postaci bomb z palącym się lontem, kranów, kapci i wielu innych przedmiotów, które warto połączyć w pary zamiast wyrzucać poza obszar układanki.

## 9 TURBO ESPRIT Durell Software

Wszechstronna firma Durell i tym razem trafiła w dziesiątkę. Symulacyjno-zręcznościowa jazda samochodem Lotus Esprit po jednym z czterech angielskich miast do dziś wzbudza niemałe emocje. Detektyw prowadzący Lotus ściga handlarzy narkotyków. Jego zadaniem jest zde maskowanie gangu przez wylapywanie podejrzanych o przestępstwo samochodów. Lotus wyposażony jest w karabin, który jest decydującym argumentem kierowcy, ale uwaga na przechodniów.

## 10 DETECTIVE A.P. Software

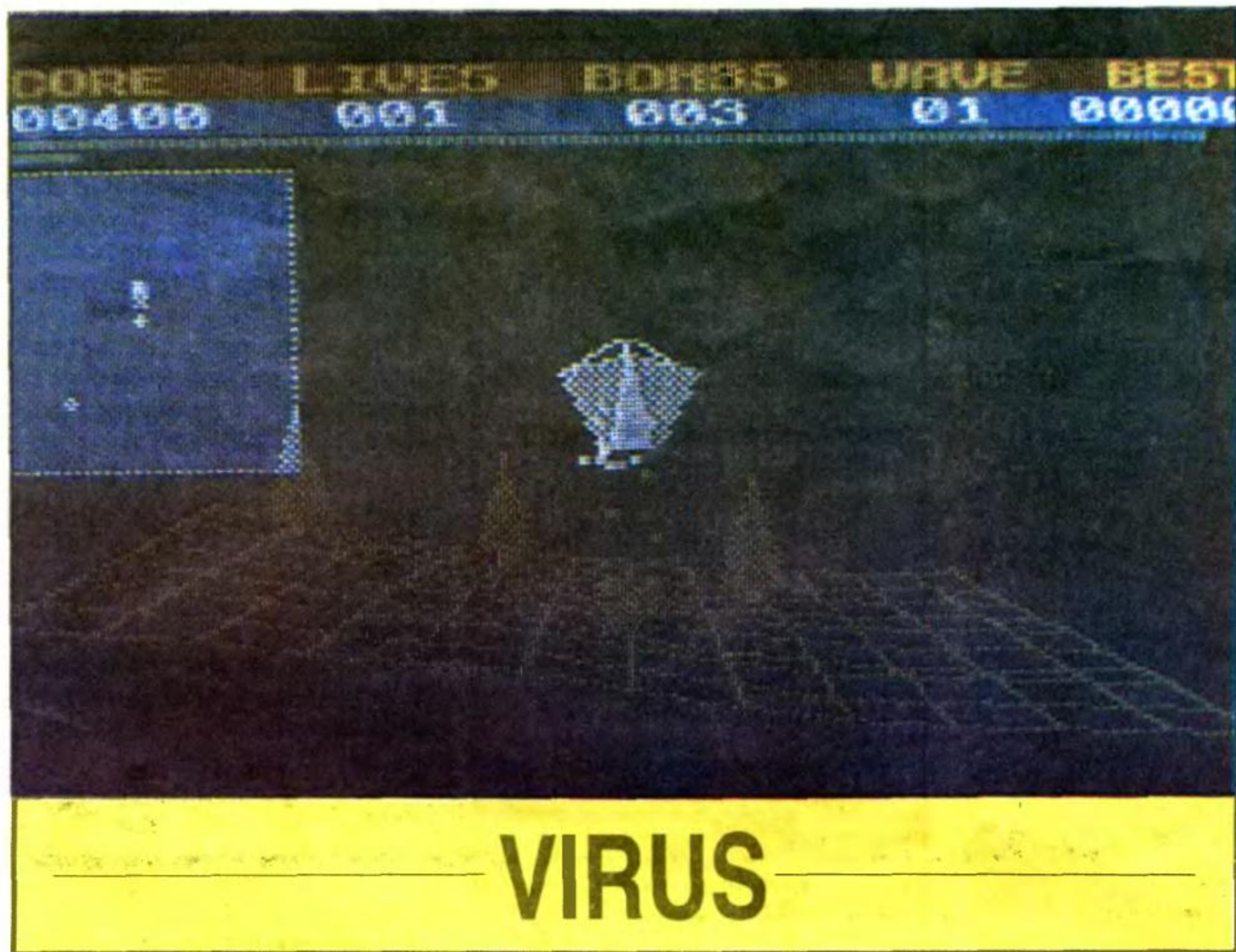
Tu wcielić się można w rolę Sherlocka Holmesa, angielskiego detektywa. Tym razem ma on zde maskować mordercę czcigodnego sir McFungusa. W domu denata znajduje się mnóstwo osób, z których każda może okazać się winna. Duża spostrzegawczość, umiejętność kojarzenia faktów i wytrwałość pomogą w wykraconiu mordercy.

Druga dziesiątka przedstawia się następująco:

11. BARBARIAN
12. RENEGADE
13. ACE
14. ANTIRIAD
15. POP EYE
16. HARDBALL
17. WEST BANK
18. GHOST CHASER
19. GUN FRIGHT
20. CHIMERA



S.O.S.



**Kraj** został zaatakowany przez kolejne fale statków kosmicznych przybyłych z obcej, wrogo usposobionej planety. Zamiast jednak atakować wojskowe instalacje obronne, obcy zdecydowali się na użycie broni biologicznej i rozpoczęli rozsiewanie czerwonego wirusa. Wirus ten jest zabójczy zarówno dla roślin jak i zwierząt oraz ludzi. Łatwo więc domyślić się, co się stanie, gdy wirus rozprzestrzeni się po całym kraju.

Gdy zawiodły już wszystkie możliwości obrony, mieszkańcy zaatakowanego kraju zdecydowali się wystąpić do walki swą najnowszą bronią — Hoverplane, za sterami którego musisz zasiąść właśnie ty. Zadanie, jakie postawiono przed tobą, jest jedno: zniszczyć obcych, nie dopuścić do rozprzestrzeniania się czerwonego wirusa i ocalenie kraju. Czym dysponujesz przystępując do walki? Jak już wspomniano, jest to Hoverplane — latający pojazd wyposażony w najnowsze zdobycze przemysłu zbrojeniowego: skaner dalekiego zasięgu, działko laserowe oraz pewną ilość bomb służących do niszczenia obcych ukazujących się na ekranie.

Gra ma doskonałą, trójwymiarową grafikę (jest to chyba najlepiej zrobiona grafika wśród gier na Spectrum dostępnych do tej pory). Autorzy nie ograniczyli się do rysowania konturów ale wypełnili je, co oczywiście polepsza znacznie walory gry. Powierzchnia ziemi przedstawiona jest jako siatka kwadratów, co ułatwia orientację w zróżnicowanym terenie. Często można spotkać rosnące na ziemi drzewa, nieźle narysowane, szczególnie w porównaniu np. z Tomahawk. Mniej więcej w środku ekranu unosi się majestatycznie Hoverplane, wykonując drobne, wahadłowe ruchy. „Okno” w lewym górnym rogu to widok skanera — pokazuje orientacyjną odległość obcych od naszego statku. Tuż nad skanerem znajdują się dwa wykresy: jeden z nich informuje o wysokości Hoverplane-u nad

powierzchnią ziemi a drugi o ilości pozostałego paliwa (można je uzupełniać lądując w macierzystych bazach). Oprócz tego z ekranu możemy się dowiedzieć o aktualnym wyniku, liczbie pozostałych „żyć”, ilości bomb, numerze fali obcych, z którymi walczymy oraz o najlepszym wyniku, uzyskanym do tej pory.

Nasz Hoverplane mimo, iż jest najnowszym krzykiem techniki, jest jednak trudny w sterowaniu. Odbywa się ono przez regulację siły ciągu oraz odpowiednim ustawieniu statku. Wymaga to pewnego treningu a najłatwiej będzie to przychodzić tym, którzy latali już na symulatorach śmigłowców (Tomahawk, Gunship), gdyż technika jest podobna. Autorzy wprowadzili jednak pewne utrudnienia: gdy Hoverplane osiągnie dużą wysokość, odcinany jest dopływ paliwa i statek zaczyna spadać.

Podczas gry możemy włączyć mapę pokazującą, jak część kraju jest już zarażona wirusem (kolor czerwony). Trzeba się więc spieszyć, aby zniszczyć wszystkie statki nieprzyjaciela, zanim zaraza pokryje cały kraj. A przeciwników mamy różnych: latające spodki rozsiewające wirus, wysoko latające bombowce a także groźne bezpośrednio dla Hoverplane statki-samobójcy. Zderzenie z którymkolwiek z nich prowadzi do rozbicia naszego statku.

Gdy uda nam się zniszczyć wszystkie statki przeciwnika, otrzymujemy premię, której wysokość zależy od nie zarażonej powierzchni kraju. Jest więc o co walczyć, gdyż dodatkowy statek otrzymujemy po zdobyciu 5000 pkt.

Podsumowując: gra należy do trudnych ale doskonała grafika i interesująca tematyka powodują, iż potrafi ona wciągnąć.

**Komputer: ZX Spectrum 48/+, Commodore 64, ATARI ST**

(mz)

## PAMIĘTNIK POKE-rzysty

Mieliśmy dość długą przerwę w publikowaniu nieśmiertelności do gier i chcemy teraz wynagrodzić ten długi czas oczekiwania. Na podstawie angielskich miesięczników „CRASH” i „Sinclair User” a także własnych doświadczeń stworzyliśmy słownik POKE-ów do gier na Spectrum. Będziemy publikować go partiami, a w przygotowaniu czeka około tysiącpozycyjny słownik do gier na Commodore. A narazie wszystko na A.

A  
Action Force II — 5145,36  
Ace — 32506,0: 32507,0: 32508,0

Ad Astra — 28591,0: 28592,0: 28593,0  
Agent X — 26099,0: 25917,0  
Agent X II — cz. 1 — 57821,0: cz. 2 — 62499,0: cz. 3 — 50561,0  
Ah Diddums — 2492,255  
Airwolf II — 53471,0  
Alien 8 — 43735,201  
Aline Highway — 39443,0: 39142,0: 35125,0  
Aliens — 31014,0: 30738,0: 34484,195  
Amoroute — 46192,0  
Amazon Women — 57590,183  
Android — 52250,32  
Arcadia — 25776,  
Arkanoid — 33702,0  
Army Moves — Cz. 1 — 54597,0: cz. 2 — 53772,0  
Athena — 50267,0: 55268,61: 51212,0  
Attack of Killer Tomatoes — 25323,0: 49433,81  
Auf Wiedersehen Monty — 42160,201: 37002,0

O co chodzi w grach TARZAN i KNIGHT LORE II na Spectrum? W zamian oddam opisy oraz mapy do gier SILENT SERVICE i RAMBO.

**Jarosław Nowak, ul. J. Bema 12, 62-500 Konin**  
Jak ukończyć grę MONTEZUMA'S REVENGE? Ile jest etapów w grze FROGGER Vo. II? Mam Atari 65 XE.

**Grzegorz Dubiel, ul. Podhalańska 36d, 34-410 Rabka**  
Jak uruchomić grę TOMAHAWK?

**Dariusz Ulikowski, ul. Ozimska 46 m 4, 45-058 Opole**  
Bardzo proszę o opisy do gier AREX, STORM, SNIPPER na Atari 800 XL.

**Szymon Pilch, ul. Jabłońskiego 73, 41-808 Zabrze 8**  
Jakie symulatory istnieją na komputer Timex 2048? Poszukuję POKE'ów do gry ROLAND RAT RACE.

**Marcin Jaskółski, ul. Boh. Monte Cassino 21 m 22, 15-888 Białystok**

Od dawna poszukuję opisów do gier EUROBUSINESS i UNIVERSAL HERO. W zamian ułatwienia do różnych gier.

**Marcin Szarek, ul. Na błoni 11a m 56, 30-147 Kraków**  
Proszę o pomoc w grach: MISSION ELEVATOR, STAR TREK, CRAFTON, SUPER HUEY — wszystkie na Atari 520 ST.

**Marisz Juskowiak, ul. Żukowa 5 m 3, 78-400 Szczecinek**

Mam Atari 800 XL. Szukam dokładnego opisu do gier: MONKEY MAGIC, BMX SIMULATOR, SPIKY HAROLD, SPELLBOUND i MOUSE TRAP.

**Anna Wengrzik, ul. Nickla 119 m 5, 41-908 Bytom**  
Nie mam ciekawych programów muzycznych i graficznych na mój Atari 65 XE. Mogę się wymienić na m.in.: NINJA, RAID OVER MOSCOU, BRUCE LEE. Mam także dobry program kopiujący, który przegrywa gry nawet w 1400 b-dów.

**Tomasz Szopa, Os. Oświecenia 3 m 34, 31-613 Kraków**

Pilnie poszukuję opisów gier PARADISE, REVERSI, ALIEN 8 na Timexa 2048. W zamian odstąpię opisy do ATIC ATAC, FIST, BOMB JACK.

**Przemysław Gaca, Os. Kosmonautów 1 m 118, 61-621 Poznań**

Szukam opisów gier HARRIER i ZAXXON oraz nieśmiertelności do nich. Komputer C-64.

**Wojciech Trzcionka, ul. Zakątek 7, 43-400 Cieszyń, woj. bielskie**

Kto przyśle mi opis gry PHOENIX — wersja na Spectrum 48?

**Grzegorz Zaczekowski, ul. Lotników 1 m 46, 78-520 Złocieniec**

Poszukuję programu użytkowego ART STUDIO w wersji polskiej. W zamian inne programy użytkowe, np. ARTIST, LEONARDO, POLSKIE LOGO.

**Michał Polak, Os. E. Plater 7c m 2, 66-620 Gubin**  
Proszę o pomoc w grach KING OF THE RING, CAVERN OF DEATH, SUPER HUEY. W zamian opisy do różnych gier.

**Maciej Meisler, ul. Świnoujska 5a, 54-313 Wrocław**  
Nie wiem, jak rozpocząć grę SPACE SHUTTLE. Szukam też nieśmiertelności do gier MONTEZUMA'S REVENGE, PIT-FALL II, SIDEWINDER. Komputer Atari 65 XE.

**Sebastian Różycki, ul. Jagiellońska 1 m 22, 03-721 Warszawa**

KRÓL I KRÓLOWA GIER



**Ania Kałasa**  
III B szk. nr 130 w Warszawie  
komputer — C-6C  
ulubione gry: Donald Duch,  
Bruce Lee  
hobby: angielski, narty



**Piotr Frenkiel**  
III kl szk. 220 w W-wie  
komputer: Amstrad 6128  
ulubione gry: Arnheim, Bomb Jack  
Cyberoid  
hobby: historia i język francuski

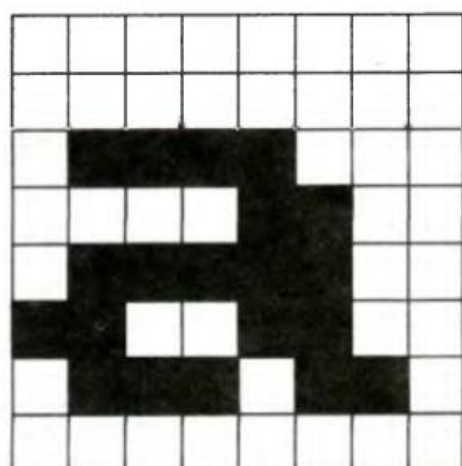


# HISOFT PASCAL

## i polskie litery

**Każdy komputer posiada zestaw znaków alfanumerycznych (liter i cyfr) oraz semigraficznych, przy pomocy których możliwa jest wizualna komunikacja między systemem i użytkownikiem.**

W większości mikrokomputerów informacja o kształcie tych znaków przechowywana jest w pamięci stałej ROM w postaci matryc. Matryca składa się najczęściej z 64 punktów (kwadrat 8x8) i dla litery „a” ma następującą postać:



Aby matrycę przechować w pamięci, dzieli się ją na wiersze. Pustym krakom przyporządkowuje się wartości 0 a pełnym wartości 1. Ponieważ każdy wiersz składa się z ośmiu elementów (o wartości 1 lub 0) można go zapisać jako bajt. W ten sposób matryca znaku „a” przyjmuje postać:

```
x1 = 00000000 = #00
x2 = 00000000 = #00
x3 = 01111000 = #78
x4 = 00001100 = #0C
x5 = 01111100 = #7C
x6 = 11001100 = #CC
x7 = 01110110 = #76
x8 = 00000000 = #00
```

i zapisuje się ją w pamięci jako ciąg bajtów: #00, #00, #78, #0C, #7C, #CC, #76, #00.

Jak można zmienić kształt znaku?

Przed wszystkim należy przepisać matryce znaków z pamięci ROM do RAM, gdyż tylko wtedy będziemy mogli je modyfikować.

W Basicu do tego celu służy instrukcja Symbol After n, natomiast w Hisoft Pascalu musimy sami zdefiniować podobną instrukcję. Skorzystamy z procedury systemowej o wektorze wywołania BBAB, która wymaga podania dwóch parametrów: numeru znaku, od którego przepisujemy matryce (przekazywanego przez rejestr DE) i adresu, pod który przepisujemy matryce do RAM-u (przekazywanego przez rejestr HL).

My przepisujemy matryce znaków o kodach powyżej 31 (tzn. pomijamy znaki sterujące) do obszaru pamięci o adresie początkowym 9300 (wartość ustalona doświadczalnie jako optymalna).

Posiadając tę informację możemy zdefiniować procedurę RamChar (patrz program). Do ustalania kształtów nowych znaków zdefiniujemy procedurę Symbol (n,x1,x2,x3,x4,x5,x6,x7,x8), która wymaga podania następujących parametrów: n — kod definiowanego znaku, x1,x2,...,x8 — kolejne bajty opisujące matrycę.

Do modyfikowania matryc użyjemy instrukcji Poke (adr, chr(x)); gdzie: adr — adres danego bajtu, x — nowa wartość bajtu.

Adres obliczamy ze wzoru:  $adr = 9300 + (n - 32) \times 8 + nr \text{ bajtu}$ , gdzie n — kod znaku, nr bajtu — numer modyfikowanego bajtu (0 + 7).

Opisane wyżej procedury możemy wykorzystać w celach praktycznych do zdefiniowania tzw. polskich liter: ą, ę itp.

W powyższym programie dō zdefiniowania polskich liter użyto kodów od 224 wzwyż. Powoduje to, że znaki nie są dostępne bezpośrednio z klawiatury, co może sprawić kłopoty. Nic nie stoi na przeszkodzie aby użytkownicy skorzystali z innych kodów. W analogiczny sposób można zdefiniować np. alfabet rosyjski. Pozostawiam to jednak czytelnikom.

*Miroslaw Rybiński*

```
Program DemoChar;
var
  a : integer;

procedure RamChar;
begin
  rDE := 32;
  rHL := #9300;
  User(BBAB);
end; { RamChar }

procedure Symbol (n, x1,x2,x3,x4,x5,x6,x7,x8 : integer);
begin
  Poke (#9300+(n-32)*8, chr(x1));
  Poke (#9300+(n-32)*8+1, chr(x2));
  Poke (#9300+(n-32)*8+2, chr(x3));
  Poke (#9300+(n-32)*8+3, chr(x4));
  Poke (#9300+(n-32)*8+4, chr(x5));
  Poke (#9300+(n-32)*8+5, chr(x6));
  Poke (#9300+(n-32)*8+6, chr(x7));
  Poke (#9300+(n-32)*8+7, chr(x8));
end;

procedure PolChar; { Procedura definiująca polskie znaki }
begin
  RamChar;
  Symbol (224, #00, #00, #78, #0C, #7C, #CC, #76, #00);
  Symbol (225, #18, #3C, #66, #66, #7E, #66, #67, #03);
  Symbol (226, #0C, #08, #3C, #66, #60, #66, #3C, #00);
  Symbol (227, #18, #3C, #66, #C0, #C0, #66, #3C, #00);
  Symbol (228, #00, #00, #3C, #66, #7E, #60, #3C, #0C);
  Symbol (229, #FE, #62, #68, #78, #68, #62, #FE, #06);
  Symbol (230, #38, #1A, #1E, #78, #58, #18, #3C, #00);
  Symbol (231, #F0, #60, #68, #78, #E2, #66, #FE, #00);
  Symbol (232, #0C, #08, #DC, #66, #66, #66, #66, #00);
  Symbol (233, #18, #D6, #E6, #F6, #DE, #CE, #C6, #00);
  Symbol (234, #0C, #08, #3C, #66, #66, #66, #3C, #00);
  Symbol (235, #0C, #38, #6C, #C6, #C6, #6C, #38, #00);
  Symbol (236, #0C, #08, #3C, #60, #3C, #06, #7C, #00);
  Symbol (237, #18, #3C, #60, #3C, #C6, #66, #3C, #00);
  Symbol (238, #18, #00, #7E, #4C, #18, #32, #7E, #00);
  Symbol (239, #30, #FE, #8C, #18, #32, #66, #FE, #00);
  Symbol (240, #0C, #08, #7E, #4C, #18, #32, #7E, #00);
  Symbol (241, #10, #FE, #AC, #18, #32, #66, #FE, #00);
end; { of Polchar }

begin { DEMO }
  PolChar;
  write ('Przyk', chr(230), 'ad:');
  write (chr(241), 'r', chr(234), 'de', chr(230), 'ko');
  writeln;
  for a := 224 to 241
    do write (chr(a), ' ');
end.
```

# DRUGIE OBLICZE

**Język Pascal został opracowany pod koniec lat 60-tych. Od tej pory minęło w informatyce kilka epok.**

Pojawiały się stopniowo komputery o coraz lepszych parametrach. Powodowało to konieczność przystosowania podstawowej wersji języka do wykorzystania nowych możliwości jakie dawał sprzęt. Poszczególne modyfikacje miały postać oddzielnych pakietów procedur dołączonych do istniejących już kompletów lub całkowicie nowych dialektów (Turbo, MT+). Większość tych modyfikacji opiera się na wykorzystaniu tzw. oprogramowania wewnętrznego (firmware) czyli zestawu procedur obsługujących tryb tekstowy graficzny, generatory dźwięku urządzenia pamięci masowych itp. Komputery serii CPC 464 posiadają

bogaty zestaw takich procedur zapisanych w pamięci ROM (od adresu 0000H do 3FFFH). Dostęp do nich umożliwiają wektory wywołań zawarte w roboczym obszarze pamięci RAM. Większość procedur została szczegółowo opisana w Bajtku (od nr 2/87) a sposób korzystania z nich, z poziomu assemblera, w nr 12/87.

Każdy użytkownik komputera zdaje sobie sprawę z potężnych możliwości jakie daje znajomość procedur systemowych i umiejętność korzystania z nich. W jaki sposób wykorzystać te możliwości z poziomu Hisoft Pascala? Dla lepszej ilustracji zajmijmy się konkretnym przykładem. Większość początkujących programistów utrudnia życie brak procedur graficznych w Pascalu. Ów brak można jednak wyeliminować.

Z Bajtka (nr 6/87) dowiadujemy się, że procedurą odpowiedzialną za wy-

świetlanie dowolnego punktu ekranu (pixela) jest procedura o wektorze wywołania BBEA (heksadecymalnie). Wymaga ona podania dwu parametrów — współrzędnych zapalonego punktu: X i Y.

Parametry te są ładowane odpowiednio do rejestrów DE i HL. Z poziomu assemblera 280 sprawa wygląda prosto: wystarczy napisać następującą procedurę:

```
Przykład I:
LD DE, 320
LD HL, 200
CALL BBFAH
RET
```

Po jej wywołaniu zostanie zapalony punkt w środku ekranu. Modyfikując zawartość rejestrów DE i HL można zapalać inne punkty.

W Hisoft Pascalu istnieje instrukcja inline, która umożliwia wstawianie w tekst programu źródłowego fragmen-

tów zapisanych w kodzie maszynowym. Wystarczy teraz przetłumaczyć mnemoniki assemblera na kod maszynowy i wstawić do instrukcji inline aby otrzymać funkcjonalny odpowiednik procedury z przykładu I.

**Przykład II:**

```
PROGRAM DemoPlot;
PROCEDURE Plot;
BEGIN
  INLINE (#11, #40, #01, #21, #C8,
    #00, #CD, #EA, #BB);
END; Plot
BEGIN
  PLOT;
END.
```

Wykonanie powyższego programu powoduje identyczny skutek jak w przykładzie I.

Pojawia się jednak zasadniczy problem: w jaki sposób zmodyfikować procedurę PLOT aby była ona uniwersalna i umożliwiała zapalenie dowol-



nych punktów ekranu a nie tylko jednego? W jaki sposób przekazywać parametry do wnętrza procedury PLOT w takiej postaci?

Rozwiązanie tego problemu może przerastać możliwości wielu początkujących programistów. Nie warto się załamywać. Otóż projektanci z firmy Hisoft rozwiązali ten problem za nas, wzorując się na zapisie assemblerowym z przykładu I. Przede wszystkim wprowadzili procedurę User (nn), która umożliwia wywoływanie m.in. procedur systemowych i jest odpowiednikiem rozkazu Call nn assemblera Z80. Drugim udogodnieniem jest wprowadzenie przez autorów kompilatora tzw. zmiennych rejestrowych. Zmienne te, jak wskazuje nazwa, reprezentują rejestry mikroprocesora. Mają one charakter zmiennych predefiniowanych, tzn. nie wymagających deklarowania przez użytkownika. Nazwy zmiennych rejestrowych składają się z dwu części: litery 'R' oraz symbolu danego rejestru: np. A, B, HL itp. rejestrowi akumulatora odpowiada zmienna R\$, rejestrowi podwójnemu HL — RHL.

Jeżeli chodzi o typ tych zmiennych, to zmienne rejestrów podwójnych są typu integer tzn. żeby załadować do rejestru BC wartość 1000 należy wykonać instrukcję: RBC := 1000;. Natomiast zmienne rejestrów pojedynczych są typu Char, tzn. żeby załadować do akumulatora wartość 13 należy wykonać: RA := chr(13); itp. Po tych informacjach napisanie uniwersalnej procedury PLOT w Hisoft Pascalu nie przedstawia większych trudności:

```
PROGRAM Demo Plot2;  
VAR A: INTEGER;  
PROCEDURE PLOT (X,Y: INTEGER);  
BEGIN  
  RDE:= X;  
  RHL:= Y;  
  USER (# BBEA);  
END; {Plot}  
BEGIN {DEMO}  
PAGE;  
FORA:= 0 TO 600 DO  
  PLOT (A,200);  
END.
```

W podobny sposób możemy definiować inne pożyteczne procedury, także o charakterze funkcji. Np:

```
PROGRAM Test Pixela;  
VAR A, B: INTEGER;  
FUNCTION TEST (X, Y: INTEGER):  
  INTEGER;  
BEGIN  
  RDE:= X;  
  RHL:= Y;  
  USER (H BBFO);  
  TEST:= ORD (RA);  
END; {TEST}  
BEGIN  
  WRITE ('X='); READ (A);  
  WRITE ('Y') : READ (B);  
  WRITE ('Punkt o współ. (x,y) ma ko-  
    lor nr, test (A,B):  
  END.
```

Program powyższy sprawdza jaki kolor ma punkt o współrzędnych X,Y. Podobne przykłady można by mnożyć. Jednak nie to jest moim celem. Chciałbym aby podane powyżej informacje stały się twórczą inspiracją dla użytkowników Hisoft Pascala i przyczyniły się do spopularyzowania tego języka.

Mirosław Rybiński

# PO CO NOWY KLAN?

## Czym różni się kalkulator od komputera i dlaczego wolimy kupić nawet najgorszy komputer niż dobry kalkulator?

Na to pytanie, jak również na wiele innych, postaram się odpowiedzieć w nowo powstałym klanie użytkowników kalkulatorów. O tym co znajdzie się w klanie przeczytacie trochę dalej, natomiast na wstępie wyjaśnię, dlaczego chcę zająć się w dobie komputerów i super komputerów właśnie kalkulatorem.

Gdy porówna się moce obliczeniowe współczesnych kalkulatorów (patrz Bajtek nr 11 z 1988 roku) z możliwościami komputerów i uwzględni się fakt, że za całe zasilanie kalkulatorom wystarcza kilka małych baterijek to porównanie nie wypada tak różowo dla komputerów. Drukarki laserowe i strumieniowe, magnetofony cyfrowe, stacje dysków, pióra służące do czytania kodu paskowego trafiły już dość dawno do kalkulatorów. Obecnie więc kalkulatory i komputery kieszonekowe to nic innego jak wysoce wyspecjalizowane komputery przeznaczone do błyskawicznego, a przynajmniej szybkiego, rozwiązywania zawiłych problemów matematycznych, prowadzenia skomplikowanych obliczeń inżynierskich i naukowych, a następnie drukowanie wyników czy zapamiętywa-

nie ich na różnego rodzaju nośnikach lub przesyłania danych przez telefon (modemy trafiły tu już też). W rzeczywistości prawdziwe komputery służą do wszystkiego, ale na ogół nie do obliczeń do, których nie są po prostu przygotowane. Przekonali się zapewne o tym Czytelnicy gdy osiągane dokładności obliczeń na komputerach 8-bitowych były rzędu  $10^{-6} \div 10^{-8}$  podczas gdy najgorszy kalkulator często osiągał dokładność o 2—3 rzędy większą. Komputery świetnie się nadają do prowadzenia i zarządzania bazami danych, są niezastąpione gdy potrzebujemy inteligentnej maszyny do pisania (edytory tekstu), można na nich generować grafikę, muzykę a od czasu do czasu nawet pograć, ale czy akurat zawsze o to nam chodziło gdy do wymarzonego komputera szepcaliśmy czule „Mój Tyś”. Jak się okazuje komputer był często powodem do rozczarowań.

Szybko pryskały marzenia o bibliotekach podprogramów, o stworzeniu oprogramowania wspomagającego pracę zawodowo itp. A kalkulator był, jest i będzie, i nie tak łatwo da się zastąpić przez komputer. Dlatego w piśmie komputerowym jakim jest Bajtek stworzyliśmy (naruszając tymczasowo) klan użytkowników kalkulatorów. Zapewniam jednocześnie, że Bajtek nie zmienia swojego profilu i komputerami zajmować się będzie. Jeżeli nasze pomysły się Wam spodobały to klan ma szansę zadomowić się w Bajtku a jeżeli nie to powrócimy wkrótce do starej postaci tematycznej. Informuję, że i użytkownicy komputerów znajdą w tej rubryce materiał do wykorzystania na swoich maszynach.

# KALKULATORY

Aby na koniec przekonać zwolenników liczenia wszystkiego na komputerze o przydatności kalkulatora przedstawię przykład jaki spotkałem w Młodym Techniku. Autor dumny był, że aby rozwiązać zadanie  $X + 2/3X - 1/3(X + 2/3X) = 10$  na komputerze potrzebował około kwadransa (włączyć komputer, znaleźć dyskietki systemową i z językiem, wczytać je, wpisać problem, nacisnąć ENTER, włączyć drukarkę, włożyć papier, wydrukować wynik, wyłączyć wszystko, schować dyskietki). Wiedziony ciekawością postanowiłem rozwiązać ten sam problem na kalkulatorze. Zajął mi to 35 sekund a z wydrukiem 40 sekund.

Czym chcemy się zajmować? Planów jest dużo jednak ich realizacja zależna jest od kalkulatorów jakie posiadacie. My nie wiemy czy są to kalkulatory czterodziałaniowe, naukowe, programowalne czy programowane. Wiemy, że na rynku są kalkulatory takich firm jak Texas Instruments, Hewlett-Packard, Casio, Sharp oraz kalkulatory radzieckie, ale czy taka jest właśnie kolejność na liście popularności... Dlatego też początkowo w klanie znajdują się informacje i odpowiedzi dotyczące kupna kalkulatorów, sprawdzania ich możliwości. Przedstawimy Wam klawiatury kalkulatorów, na które z tego czy innego powodu warto zwrócić uwagę. Jednocześnie czekamy na Wasze listy, które pomogą nam dostosować materiał klanowy do Waszych potrzeb. Zapraszamy do wspólnego redagowania klanu, prosimy o nadsyłanie własnych materiałów. Mile widziana jest również konstruktywna krytyka.

W klanie pokażemy metody obliczania całek na kalkulatorach czterodziałaniowych, pokażemy jak wykorzystać w pełni statystykę kalkulatorową, ujawnimy jak otrzymać na kalkulatorze wynik rzędu  $10^{300}$ , podpowiemy jak ułatwiać sobie prowadzenie żmudnych obliczeń na kalkulatorach naukowych a od czasu do czasu zaprezentujemy jakąś grę napisaną dla najprostszych kalkulatorów programowalnych. Mile widzimy gry pisane przez Was. Czekamy na listy.

Dominik Falkowski

## Firma Hewlett-Packard analizując wydajność pracy informatyka doszła do ciekawego spostrzeżenia.

Otóż często podczas uruchamiania programów w kodzie maszynowym zachodzi potrzeba dokonania prostych obliczeń w celu usunięcia błędów. Nie byłoby to zadanie trudne gdyby nie to, że informatyka posługuje się przeważnie systemami liczenia o podstawie różnej od dziesięciu: binarnym, oktalnym i heksadecymalnym, co znakomicie utrudnia prowadzenie prostych działań arytmetycznych. Problemem jest czasami bogatszy repertuar funkcji zawierający oprócz arytmetycznych również działania logiczne.

Firma postanowiła wyjść naprzeciw zapotrzebowaniu i opracowała kalkulator informatyczny model HP-16C. Jego możliwości jak na tak małe wymiary są bardzo duże. Oczywiście może on pracować w jednym z czterech powszechnie stosowanych systemów liczenia, dokonywać konwersji między tymi systemami, wykonywać działania arytmetyczne w zakresie kalkulatora czterodziałaniowego oraz szereg operacji logicznych NOT, OR, AND, XOR. Szczególnie szeroki jest zasób operacji binarnych. Obejmuje on: przesunięcia, rotację, rotację z przeniesieniem, ustawianie, maskowanie i testowanie bitów jak również generuje sumy kontrolne. Wszystkie operacje wykonywane są na słowie 64



bitowym choć jego długość może zostać zmniejszona przez użytkownika np. do 32 bitów. Kalkulator jest w pełni programowalny (do 203 linii programu), co przyczynia się do poszerzenia jego możliwości. Programowanie można stosować przy wielokrotnym testowaniu segmentów programów w kodzie maszynowym poprzez symulację ich pracy programem kalkulatora. HP-16C posiada również szereg funkcji ułatwiających edycję programu, umożliwiających określenie, w jakim systemie pracuje kalkulator oraz przeglądania liczb, których długość prze-

kracza możliwości wyświetlacza. Lite-ra C w nazwie podobnie jak wielu innych kalkulatorach tej czy innej firmy oznacza, że po wyłączeniu kalkulatora wszystkie informacje raz wpisane pozostają niezmiennione i są gotowe do ponownego ich użycia. Wszystkie te udogodnienia czynią z tego kalkulatora narzędzie niezwykle przydatne podczas pracy przy klawiaturze komputera, kiedy to stajemy przed koniecznością wykonania prostego działania \$A8 AND %1011 OR +3 bez zmiany zawartości ekranu komputera.

A.Z.

# KALKULATOR DLA INFORMATYKA



## HISTORIA LUBI SIĘ POWTARZAĆ

Minikalkulator znają wszyscy, jest on jednym z urządzeń powszechnego użytku. Mało kto uświadamia sobie, że był to pierwszy przedstawiciel techniki cyfrowej jaki zawita w naszym domu.

Jeszcze zupełnie niedawno elektroniczne maszyny liczące były dostępne tylko dla wąskiego kręgu osób związanych bezpośrednio z matematyką. W biurach, w kasach i szkołach królowały niepodzielnie coraz bardziej wyrafinowane kręciółki i coraz dłuższe i szersze suwaki logarytmiczne. Niektóre pracownice konstrukcyjne chlubiły się posiadaniem elektronicznych liczydeł, które mając wymiary maszyny do pisania były w stanie biegle dodawać i odejmować, mnożyć i dzielić pełniąc jednocześnie z racji swojej temperatury rolę kaloryfera.

Dziś te czasy należą już do przeszłości. Wraz z rozwojem mikroelektroniki na świecie pojawiły się minikalkulatory. Małe, mieszczące się w kieszeni, zadowolające się bateryjnym zasilaniem, o mocy obliczeniowej niejednokrotnie przewyższającej moc komputerów minionych generacji. Początkowo wykonywały tylko cztery podstawowe działania, następnie nauczyły się potęgować, pierwiastkować oraz obliczać procenty. Wkrótce otrzymały też bardzo przydatny mechanizm jakim była pamięć, która uprościła wykonywanie skomplikowanych obliczeń i ułatwiła ich wielokrotne powtarzanie.

Dla niektórych modeli rozwój zakończył się już na tym etapie. Okazało się, że dla pani domu, pierwszoklasisty, czy też klienta PKO są one odpowiednio i całkowicie wystarczające. Pozostało tylko zmieniać formę pozostawiając wypróbowaną treść i w ten sposób kalkulatory zawędrowały do puderniczek, zegarków, stały się tak cienkie jak kartka papieru lub też tak mocne, by nie dał im rady uparty przedszkolak.

Równoległe trwał wyścig o coraz to większą moc obliczeniową. Kolejnym etapem była propozycja kalkulatora dla ucznia szkoły średniej i inżyniera. Posiadał on już podstawowy zestaw funkcji matematycznych, aby niezależnie się od tablic, oraz odpowiednią dokładność, która zapewniała przeprowadzenie nawet bardzo skomplikowanych obliczeń. Aby zbliżyć sposób posługiwania się kalkulatorem do przepisywania wzorów z książki zaopatrzone go w możliwość korzystania z kilkupoziomowych nawiasów i umiejętność prowadzenia obliczeń w ogólnie przyjętej kolejności (np. mnożenie przed dodawaniem). W ten sposób narodziła się klasa kalkulatorów inżynierskich, która posiadając znaczną przewagę nad kalkulatorami czterodziałaniowymi nie była już tak spójna, co powodowało pewne kłopoty przy posługiwaniu się nieznanym modelem nawet tego samego producenta.

Powstanie tej klasy wiąże się z jeszcze jednym ważnym faktem: powstała grupa kalkulatorów o zupełnie zmienionej filozofii prowadzenia obliczeń tzw. Odwrotna Notacja Polska. Był to system prowadzenia obliczeń opracowany przez naszego rodaka i oparty o stos dostępny dla użytkownika, ale nie korzystający z nawiasów i znaku równości. (Bliżej o Odwrotnej Notacji Polskiej już wkrótce).

Kolejny etap rozwoju podyktowany był koniecznością ułatwienia wielokrotnego powtarzania cykli obliczeń. Do rozwiązania tego problemu prowadziły dwie drogi. Można było przystosować kalkulator do zapamiętywania sekwencji wciskanych klawiszy, a następnie jej wielokrotnego powtarzania, lub rozszerzyć repertuar funkcji kalkulatora o najczęściej wykonywane procedury obliczeniowe. Każdy ze sposobów posiadał swoje wady. Pierwszy z nich wymagał od użytkownika pewnej wprawy związanej z uczeniem kalkulatora rozwiązywania określonego problemu. Powodował też stratę czasu przy każdorazowej zmianie rozwiązywanego problemu, związaną go z opracowaniem nowego algorytmu i wprowadzeniem go do pamięci kalkulatora. Drugi ograniczał znacznie uniwersalność i powodował skomplikowanie kalkulatora uniemożliwiające pełne jego wykorzystanie bez posłużenia się niejednokrotnie kilkutomową instrukcją obsługi.

Brak zdecydowanej przewagi którejś z grup spowodował ich równoległy rozwój. Pierwsza rozwinęła się jako kalkulatory programowane zapożyczając wiele rozwiązań z dorosłych komputerów umożliwiających np. przechowywanie programów i danych na nośnikach magnetycznych lub też podejmowanie przez program decyzji co do dalszego trybu obliczeń na podstawie wyników pośrednich. Druga grupa po początkowych roszczeniach do opanowania całego rynku przeprowadziła udane natarcie na użytkowników specjalistycznych. W ten sposób powstały kalkulatory nawigacyjne, astro-nomiczne, finansowe, fizyczne i wiele innych. Tego typu specjalizacja umożliwiła lepsze dostosowanie do potrzeb bez konieczności wbudowywania zbędnych w danej problematyce procedur.

W ten sposób dojrzałyśmy do czasów współczesnych. Obecnie na rynku kalkulatorów daje się zauważyć tendencja do zażegnania powstałego rozłamu. Usiłuje się łączyć możliwość programowania z dużymi zbiorami funkcji specjalnych znajdujących się w modułach, które można wymieniać dostosowując kalkulator do różnych wymagań. W zakresie metod programowania odchodzi się od języków typu klawiaturowego na rzecz języków wyższego rzędu np. Pascala, starając się dostosować komfort współpracy z kalkulatorem do poziomu programów komputerowych wprowadzając pracę dialogową i ekran graficzny. Ostatnią nowością jest możliwość prowadzenia obliczeń symbolicznych wraz z całkowaniem i różniczkowaniem.

Prognozy co do dalszego rozwoju kalkulatorów są trudne do określenia. W obecnych warunkach rozwój komputerów osobistych i ich wersji przenośnych stawia ostre warunki dla przyszłych kalkulatorów i należy się spodziewać powstania zupełnie nowych rozwiązań tego elektronicznego liczydła.

Adam Zakrzewski

# TEST PLOTERA PRN C41 firmy Sony

PLOTER DLA KAŻDEGO!

Parę miesięcy temu mój znajomy zwrócił mi uwagę na wymieniony w tytule ploter firmy Sony oferowany przez Centralną Składnicę Harcerską. Osobiście widziałem to urządzenie wcześniej w sklepie na Mokotowskiej w Warszawie, ale napis MSX i brak jakiejkolwiek informacji czy reklamy był przyczyną braku mojego zainteresowania tym ploterem. Dopiero, kiedy znacznie później otrzymałem ksero instrukcji firmowej [1] stwierdziłem z pewnym zdziwieniem, że jest to bardzo ciekawy sprzęt, który daje się podłączyć do każdego prawie komputera i że dwa podstawowe problemy związane z możliwością korzystania z plotera dotyczą właściwego kabelka i odpowiedniego oprogramowania.

Porównanie danych technicznych z Rolandem [2] wypadło dość ciekawie. Ploter firmy Sony jest 4 razy wolniejszy, 4 razy mniej dokładny, ma ubogie oprogramowanie, ale zajmuje znacznie mniej miejsca i kosztuje prawie 10 razy mniej. Obecna cena wynosi 460 000 zł. Dokładniejszy opis sprzętu poniżej.

## HARDWARE

Ploter waży 1.3 kg, ma rozmiary 310\*67\*108 i umożliwia sporządzanie kolorowych rysunków i wydruków na pojedynczej kartce formatu A4 lub na rolce ciągłego papieru. Dzięki mechanizmowi przypominającemu magazynek pistoletu COLT dostępny są jednocześnie cztery piórka: czarne, niebieskie, zielone i czerwone. Rysowanie wzdłuż osi X odbywa się przy pomocy poruszającego się na szynie wagonika z piórkami, a ruch wzdłuż osi Y zapewnia przesuw samego papieru. Rozdzielczość plotera wynosi 0.2 mm, a szybkość kreślenia wzdłuż każdej z osi 57 mm/s. Powtarzalność jest mniejsza niż rozdzielczość, czyli mniej niż 0.2 mm, a dokładność rzędu 1%. Pulpit sterujący urządzeniem składa się z 4 kolorowych przycisków. Dwa szare, z białymi trójkątami, służą do przesuwu papieru do przodu i do tyłu. Fioletowy klawisz pozwala zmieniać ręcznie kolor piórka, natomiast czerwony przycisk RESET umożliwia inicjalizację plotera. Dodatkowo na płycie czołowej znajduje się wyłącznik niskonapięciowego zasilania. Całe urządzenie zasilane jest poprzez zewnętrzny transformator 220V/10V, który nie ma oddzielnego wyłącznika. Wymiana informacji z komputerem odbywa się poprzez uproszczoną wersję typowego złącza Centronics.

Sam mechanizm plotera jest dość delikatny. Dotyczy to szczególnie mocowania piórek. Ta część urządzenia jest bardzo podatna na uszkodzenia podczas wymiany piórek. Podnoszenie dźwignienki dociskającej piśak odbywa się w dwóch fazach. Najpierw należy pociągać do siebie, a dopiero później podnieść. W instrukcji technicznej nie jest to wyraźnie podkreślone i pominięcie fazy pierwszej było przy-

czyną uszkodzenia udostępnionego redakcji urządzenia.

Ploter udało się częściowo naprawić, tak że mogliśmy kontynuować redakcyjny test, ale przywrócenie stanu pierwotnego wymagać będzie wymiany prawdopodobnie całego mechanizmu dociskowego. Jak poinformowano nas w CSH, serwis tego sprzętu prowadzi firma Wektor (tel. 6107244), która dysponuje częściami zapasowymi.

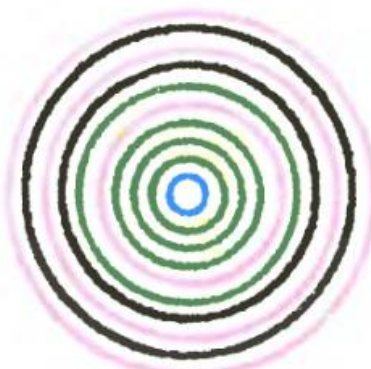
## PISAKI

Dostępne są dwa rodzaje pisaków w czterech kolorach. Są to piórka kulkowe, dające bardzo cienką i precyzyjną linię, oraz mazaki, które piszą wyraźniej ale grubiej. Każde z tych piórek da się rozebrać i jest w pewnym sensie regenerowalne. Ma to istotne znaczenie eksploatacyjne, ponieważ piórka są drogie (7600 zł — komplet), małe i prawdopodobnie na długo nie starczą.

## SOFTWARE

Oprogramowanie firmowe plotera jest dość ubogie i obejmuje umiejętność kreślenia linii prostych oraz drukowania w 16 rozmiarach i 4 kierunkach 252 znaków. Urządzenie pracuje w trybie tekstowym i graficznym. Dwa autotesty uruchamiane przez jednoczesne wciśnięcie klawisza zmiany koloru (CHANGE COLOR) razem z jednym z klawiszy przesuwu papieru demonstrują możliwości tych trybów. Autotest 1 (rys. 2) drukuje zestaw znaków standardu MSX. Autotest 2 (rys. 3) kreśli cztery kolorowe kwadraty.

Programowanie plotera jest proste i polega na wysyłaniu przez komputer, tak jak na drukarkę, od-



Rys. 1 Wynik działania programu CircleDemo



powiednich kodów sterujących. Sekwencje te zestawiono na rys. 4. Przyjęta konwencja zapisu pozwala na bezpośrednie ich zastosowanie we własnych programach pascalo- wych. Przy pracy w Basic'u konieczne są niewielkie zmiany.

Przykład 1. Przejście do trybu graficznego

```
WRITELN (LST, #27' #');
(* PASCAL *)
LPRINT CHR$(27); #';
(* BASIC *)
```

Przykład 2. Linia z pktu aktualnego do pktu x, y

```
WRITELN (LST, 'D', x', ', y);
(* PASCAL *)
LPRINT "D"; x; ", "; y;
(* BASIC *)
```

Przeglądając listę dostępnych kodów sterujących możemy stwierdzić, że brakuje w niej możliwości definiowania własnych znaków, co jest ważne, jeśli chcemy wprowadzić polskie litery. Kolejnym nierozwiązanym problemem jest kreślenie bardziej skomplikowanych krzywych takich jak okręgi, elipsy itp. Listing 1 przedstawia krótki program demonstracyjny zawierający procedurę CIRCLE, która rysuje okręgi o promieniu r. Jest to przerobiona wersja przykładowej procedury OKRĄG podanej w artykule „Algoritmy generacji krzywych na płaszczyźnie” [3]. Modyfikacja jest prosta i polega tylko na dodaniu trzech instrukcji WRITELN z odpowiednimi parametrami. Implementacja pozostałych procedur przedstawionych w [3] jest równie łatwa.

Program uruchomiono na komputerze IBM PC z ploterem podłączonym do drugiego złącza Centronics — port LPT2. W przypadku korzystania ze złącza pierwszego lub innego komputera, należy zastąpić w programie wszystkie wystąpienia pliku f przez LST i pominąć instrukcje ASSIGN, REWRITE i CLOSE. Wynik działania programu przedstawiono na rys. 1.

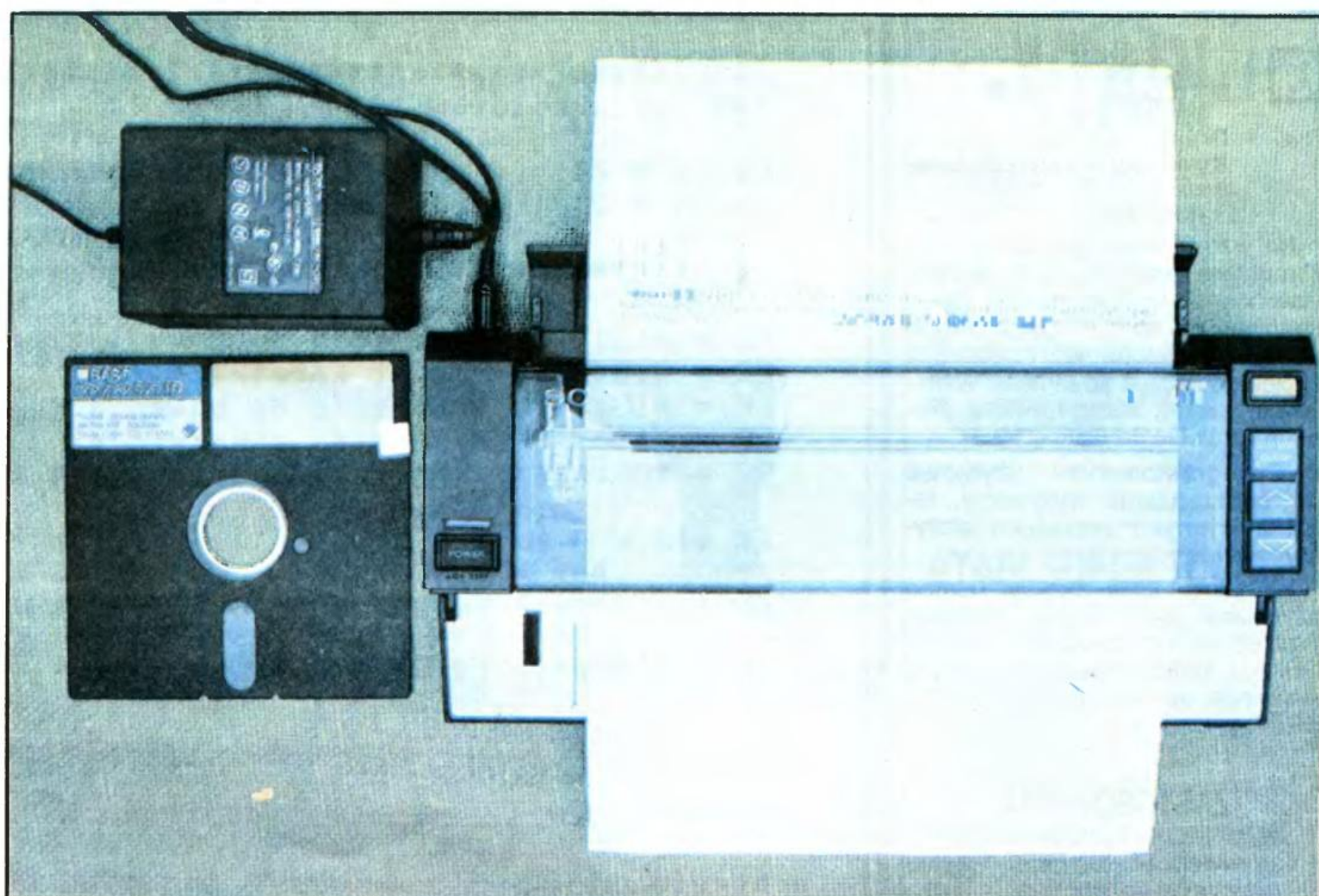
## KOMPUTERY

Prezentowany ploter przetestowano na komputerze IBM PC XT i na Amstradzie PCW 8256 wyposażonym w interface brytyjskiej firmy SCA. Ponieważ ploter wykorzystuje standardowe złącze typu Centronics, możliwe jest jego podłączenie do każdego komputera wyposażonego w to złącze. Poniżej przedyskutowano współpracę plotera z popularnym u nas sprzętem.

Jedynym komputerem nie wymagającym przeróbki jest SpectraVideo w standardzie MSX'a, dla którego wg instrukcji ploter jest przeznaczony.

1. *Komputery zgodne z IBM PC (XT, AT, Amstrad PC 1512, 1640 itp).* Komputer użyty do testu zawierał dwa złącza Centronics. Jest to typowa konfiguracja wyposażona w kartę Hercules z wyjściem na drukarkę i kartę DISK I/O, na której znajduje się drugi port równoległy. Do portu pierwszego była przyłączona zwykła drukarka, natomiast ploter podłączono do drugiego portu. Jest to rozwiązanie bardzo wygodne do pracy.

Przykład programowania plotera



w Turbo Pascalu podano w programie CircleDemo (listing 1). Jeśli chcemy korzystać z Basic'a, postępując się portem 1, możemy uruchomić krótki program w Basic'u przedstawiony na listingu 2. Zamieni on logicznie dwa porty. Pozwoli to także na tekstowy zrzut ekranu na ploter i pracę z Pascalem przy pomocy standardowego pliku LST. Zbiory tekstowe mogą być drukowane także przez użycie instrukcji COPY, np. *COPY nazwa zbioru LPT2* (lub LPT1 jeśli zamieniliśmy porty miejscami).

Uruchomienie plotera wymagało niewielkiej przeróbki typowego kabla od drukarki. Ponieważ ploter nie wystawia sygnału PAPER END (opis złącza — rys. 5) konieczne było uziemienie przez opornik 1kΩ tego sygnału (przewód nr 12) od strony komputera.

2. *Amstrad PCW 8256/8512*

Był to drugi komputer, do którego podłączono ploter. Konieczne było wyposażenie Amstrada w dodatkowe złącze Centronics. Zrobiono to przy pomocy układu firmy SCA, zgodnego z interface'm CPS 8256 [4] firmy Amstrad. Zastosowano typowy kabel od drukarki zakończony dwustronnie wtykami typu Amphe- nol. Zestaw funkcjonował bez żadnych problemów.

3. *Amstrad CPC 464/664/6128*

Komputery tego typu wyposażone są w standardowe złącze Centronics. Niestety jego przekleństwem jest dostępność tylko siedmiu bitów. Utrudnia to częściowo korzystanie z plotera, ponieważ nie możemy wydrukować znaków o kodach większych od 128. Rozwiązanie tego problemu, czyli dodanie ósmego bitu do złącza zostało omówione w [5,6]. W [6] obiecano przedstawić wersję pracującą pod CP/M'em.

Podłączenie plotera wymaga zastosowania takiego samego kabla

jak dla PCW. Programowanie możliwe jest w Basic'u pod AMSDOS'em, a także w Turbo Pascalu pod CP/M'em i w innych językach.

4. *Sinclair ZX Spectrum*

Podłączenie plotera do popularnego Spectra wymaga przede wszystkim nabycia interface'u typu Centronics. W CSH w sklepie na Mokotowskiej za około 40 tys. zł można kupić takie złącze. Potrzebny jest standardowy kabel. Programowanie w Basic'u, w HiSoft Pascalu i innych językach.

5. *Atari (8-bitowe)*

W przypadku tego komputera konieczne jest także nabycie lub wykonanie samodzielnego interface'u Centronics. W [7] podano przykładową jego realizację. Na giełdzie złącze to było dostępne za 35 tys. zł (X,88). Kabel i oprogramowanie podobnie jak dla innych 8-bitowych komputerów.

## (CIEKAWY) PROBLEMY DO ROZWIĄZANIA

Jeśli chcemy wszechstronnie i efektywnie wykorzystywać na co dzień prezentowany ploter, musimy, ze względu na jego ubogie oprogramowanie firmowe, rozwiązać wiele problemów. Jest to szerokie pole do popisu dla samodzielnych programistów, firm produkujących oprogramowanie, czy nawet Czytelników. Sądzę, że każdy Klan w Bajtku mógłby prezentować krótkie użyteczne programy na ten ploter. Poniżej przedstawiono kilka typowych zagadnień, którymi można się zająć.

### ● Polskie znaki.

W trybie tekstowym polskie znaki można wprowadzić w prosty sposób postępując się funkcją BACKSPACE, tzn. drukować np. literę /, następnie cofnąć się i dodrukować poprzeczną kreskę. W trybie grafi-

cznym plotera konieczne byłoby napisanie programu, który rysowałby dowolne znaki składając je z kolejnych kawałków krótkich linii prostych.

### ● Program do projektowania własnych znaków.

Program ten służyłby do wygodnego tworzenia własnych znaków poprzez określenie sekwencji sterujących, które prowadzą do narysowania danej litery lub symbolu.

### ● Kolorowy edytor graficzno-tekstowy.

Opisywany ploter może być wykorzystany do drukowania dowolnych tekstów w kilku kolorach i rozmiarach. Zmiana czcionek (otrzymanych np. dzięki poprzedniemu programowi) i możliwość dodania rysunków do tekstu pozwoliłaby dostarczyć użytkownikowi bardzo ciekawy i atrakcyjny program.

### ● Współpraca z gotowymi programami graficznymi (DR. DRAW, MicroDraft, AUTOCAD i inne).

Poważny problem napotkamy, jeśli zechcemy korzystać z plotera przy pracy z gotowymi programami graficznymi np. DR. DRAW, DR. GRAPH, MicroDraft, czy AUTOCAD. Ploter PRN C41 nie posiada w swoim ROMie interpretera języka HPGL (Hewlett Packard Graphics Language), a programy te wykorzystują ten standard.

Możliwe są dwa rozwiązania. Jednym z nich jest napisanie interpretera języka HPGL np. w Turbo Pascalu lub Basic'u w sposób ogólny, pozwalający na jego stosowanie na dowolnym komputerze. Drugie rozwiązanie jest mniej uniwersalne i zakłada konieczność napisania dla każdego programu własnej procedury sterowania ploterem (ang. driver).

### ● Oprogramowanie do kreślenia dowolnych krzywych.



Rys. 2 Pierwszy autotest plotera. Zestaw dostępnych znaków.





Rys. 3 Drugi autotest plotera. Kwadraty w czterech kolorach.

Na wzór pakietu graficznego dla komputera Amstrad PCW, przedstawionego w [3,8], można skonstruować taki sam zestaw procedur na każdy komputer, który będziemy chcieli podłączyć do plotera. Wskazówkę jak to zrobić zawiera program CircleDemo (listing 1).

## ● Oprogramowanie użytkowe do sporządzania wykresów, tabel i innych zestawień statystycznych.

Programy takie byłyby bardzo użyteczne przy pracy biurowej. Wzorem takiego programu jest DR GRAPH, funkcjonujący pod kontrolą systemów operacyjnych MS-DOS i CP/M.

## ZASTOSOWANIA

Jak widać z powyższego opisu prezentowany ploter jest bardzo ciekawym urządzeniem, które może znaleźć wiele interesujących praktycznych zastosowań.

Na pierwszy plan wysuwa się olbrzymie znaczenie dydaktyczne plotera. Prostota jego obsługi pozwala wykorzystywać go do nauczania elementów informatyki. Każda szkolna pracownia wyposażona w dowolne komputery mogłaby się nim posługiwać celem uatrakcyjnienia zajęć.

Indywidualny użytkownik noszący się z zamiarem kupna drukarki, może wykorzystać ploter jako wolną, ale kolorową drukarkę, w przybliżeniu dwukrotnie tańszą od typowej.

Bardzo interesujący zestaw tworzy Amstrad PCW i wspomniany ploter wykorzystujący program MicroDraft. Jest to bardzo tania alternatywa dla małych zakładów rzemieślniczych, które w swojej pracy muszą się posługiwać programami typu AUTOCAD, a czekanie na wydruk z drukarki jest zbyt długie. W tym wypadku konieczne jest napisanie nowego programu MPLOT.COM pozwalającego na wydruk rysunków generowanych przez program MDRAFT.COM. Dokumentacja pakietu MicroDraft zawiera informacje niezbędne do osiągnięcia tego celu.

Innym ciekawym zastosowaniem plotera może być wypełnianie sztywnych formularzy i dokumentów w różnych urzędach. Standardowe drukarki, zawijające papier na wałek, nie pozwalają na to. Zastosowanie wspomnianego plotera podniosłoby istotnie atrakcyjność wielu oferowanych do tej pory na rynku systemów wspomagających pracę urzędników.

## PODSUMOWANIE

Biorąc pod uwagę opisane możliwości testowanego plotera, jego parametry techniczne i cenę, uważam że jest to bardzo dobry produkt, który można polecić wielu użytkownikom.

Na marginesie chciałbym dodać, że jest to jedno z kilku ciekawych i względnie tanich urządzeń oferowanych przez CSH. Niestety zero-we zainteresowanie ich promocją

```

(*****)
(** Kody kolorow **)

black = 0: (* czarny *)    blue  = 1: (* niebieski *)
green = 2: (* zielony *)   red    = 3: (* czerwony *)

(***** Kody trybu tekstowego *****)

FF = #12: (* przejście do nowej strony *)
CR = #13: (* powrót karetki *)
LF = #10: (* przejście do nowej linii *)
LB = #11: (* powrót do poprzedniej linii *)
BS = #08: (* cofnięcie o jeden znak *)

(* #18,n - zmiana skali pisma, gdzie n = 0..15 *)

(* #27'C',n - zmiana koloru pisma, gdzie n = 0..3 *)

IG = #27'#': (* przejście do trybu graficznego *)

(***** Kody trybu graficznego *****)

OG      = 'A': (* powrót do trybu tekstowego *)

origin  = 'I': (* wybranie początku układu wsp.
                w aktualnej pozycji pisaka *)

home     = 'H': (* powrót do początku układu *)
nline    = 'F': (* przejście do następnej linii *)

(* 'C',n - zmiana pisma, gdzie n = 0..3 *)
(* 'S',n - zmiana skali pisma, gdzie n = 0..15 *)
(* 'L',n - zmiana typu linii, gdzie n = 0..15 *)
(* 'Q',n - zmiana kierunku pisma, gdzie n = 0..3 *)

(** Kody kierunkow **)

right = 0: (* na prawo *)    down  = 1: (* na dol *)
left  = 2: (* na lewo *)    up     = 3: (* na gore *)

(* 'P',napis - wydrukowanie napisu 'napis' *)

(* 'D',x,',',y - linia do pktu x,y z pkt akt.
                ang. DRAW *)
(* 'J',DX,',',DY - linia o DX,XY z pkt akt.
                ang. relative DRAW *)
(* 'M',x,',',y - przesunięcie pisma do pktu x,y
                ang. MOVE *)
(* 'R',DX,',',DY - przesunięcie względne o DX,DY
                ang. relative MOVE *)

(*****)
    
```

Rys. 4 Kody sterujące plotera PRN C41 w konwencji pascalowej.

Nr przewodu	Nazwa sygnału	Nr przewodu	Nazwa sygnału
1	STB'	19	GND
2	DATA 1	20	GND
3	DATA 2	21	GND
4	DATA 3	22	GND
5	DATA 4	23	GND
6	DATA 5	24	GND
7	DATA 6	25	GND
8	DATA 7	26	GND
9	DATA 8	27	GND
10	ACK'	28	GND
11	BUSY	29	GND
12	NC	30	NC
13	PULL UP (4.7K)	31	NC
14	GND	32	NC
15	NC	33	NC
16	GND	34	NC
17	GND	35	NC
18	NC	36	NC

Rys. 5 Opis złącza Centronics dla plotera PRN C41. GND — masa sygnałowa, NC — niepodłączona, apostrof — negacja sygnału.



```

program CircleDemo:
    (*****
    (* (C) JM Styczen 1989 *)
    (*****
var
    f : text; i : integer;

procedure Circle (r : integer);
var
    x,y, dx,dy, fa,fx,fy : integer;
begin
    x := r; y := 0; fa := 0;
    writeln (f,'R',r,',',0'); (* przesun pióro do (r,0) *)
    repeat
        if x>0 then begin dy := -1; fy := fa - 2*y + 1 end
        else begin dy := 1; fy := fa + 2*y + 1 end;
        if y>0 then begin dx := 1; fx := fa + 2*x + 1 end
        else begin dx := -1; fx := fa - 2*x + 1 end;
        if abs(fx)<abs(fy)
        then begin
            fa := fx; x := x + dx; writeln (f,'J',dx,',',0');
        end
        else begin
            fa := fy; y := y + dy; writeln (f,'J0,',dy);
        end;
    until (x=r) and (y=0);
end; (* circle *)

begin

    assign(f,'lpt2'); rewrite (f);

    writeln(f);
    writeln(f,'#27'#'); (* włącz grafike *)

    writeln (f,'M100,-100'); (* przesun pióro do pkt (100,-100) *)
    writeln (f,'I'); (* nowy początek układu wsp. *)

    for i := 1 to 20
    do begin
        writeln (f,'H'); (* przesun pióro do pocz. układu *)
        writeln (f,'C',random(4)); (* wybierz losowo pióro *)
        Circle (4 * i);
    end;

    writeln (f,'A'); close (f); (* wyłącz grafike *)

end.

```

Listing 1. Program CircleDemo.

```

10 DEF SEG = 64
20 X = PEEK (8)
30 Y = PEEK (9)
40 POKE (8), PEEK (10)
50 POKE (9), PEEK (11)
60 POKE (10), X
70 POKE (11), Y
80 SYSTEM

```

Listing 2. Program zamiany wyjść LPT1 i LPT2 na IBM PC.

na rynku wykazywane przez instytucję sprzedającą jest przyczyną niewielkiego ich rozpowszechnienia. Brak informacji technicznej z prawdziwego zdarzenia czy reklamy powinien budzić zdziwienie nawet na naszym rynku. Plotery firmy Sony sprzedawane są w sklepach CSH w tempie kilku sztuk rocznie. Niewiele chyba osób wie, że można podłączyć je nie tylko do MSX'a. Mam nadzieję, że powyższy test

prezentowany na łamach „Bajtek” ma szansę zmienić częściowo tę sytuację. Jeśli osoby zajmujące się marketingiem produktów oferowanych przez Centralną Składnicę Harcerską udostępnią redakcji opisywany ploter na dłużej, to mamy zamiar kontynuować serię artykułów poświęconych praktycznym zastosowaniom plotera PRN C41 firmy Sony w pracy z popularnymi komputerami dostępnymi u nas na rynku. Na początek zamierzamy przedstawić między innymi implementację polskich liter, „mały pakiet graficzny”, proste programy do „szkolnych” wykresów badania zmienności funkcji, itp.

Jonasz Mayer

#### Wady

- delikatna konstrukcja
- nietypowe, małe pisaki
- brak oprogramowania graficznego np. języka HPGL

#### Zalety

- + niska cena

- + standartowy interface typu Centronics
- + prostota obsługi
- + małe rozmiary

#### LITERATURA

- [1] Instrukcja firmowa do plotera Sony PRN C41
- [2] Instrukcja firmowa do plotera Roland DXY 880
- [3] J. Młodzki, „Algorytmy generacji krzywych...”, Bajtek 7/88
- [4] „CPS 8256”, Bajtek 6/88
- [5] „8 bit do CPC 6128”, Komputer 7/86, 11/87
- [6] L. Łobocki „Amstrad CPC 6128 na serio”, Komputer 11/88
- [7] M. Renner, „Centronics dla Atari”, Bajtek 11/88
- [8] J. Młodzki, „Mały pakiet graficzny...”, cz. I, II, III, Bajtek 8, 9, 12/88

Jonasz Mayer

Testowany ploter otrzymaliśmy dzięki uprzejmości dyrektora Centralnej Składnicy Harcerskiej pana W. Szantera. Dziękujemy.

# -ATARI-

Szeroki wybór oprogramowania na kasetach i dyskietkach

- co piąty program bezpłatnie
  - gwarancja jakości
  - rachunki
  - katalogi gratis
- ATR-SOFTWARE  
66-542 Zwierzyń P-1

D-158

## ATARI SUPER TURBO

udoskonalony system transmisji danych do magnetofonów firmowych i niestandardowych

- 10 razy szybciej
- 0% błędów
- 100% pewności
- współpracuje z każdym magnetofonem
- możliwość nadawania nazw zbiorom zapisanym na taśmie
- gęstość zapisu do 30 programów na jednej stronie kasy C-60

### TWÓJ MAGNETOFON JAK STACJA DYSKÓW!

Zgłoszenia: Jerzy Pietrykowski,  
tel. 37-49-64 Warszawa  
Ponadto oferujemy bogaty wybór oprogramowania.  
Wykonujemy usługi zapisywania i kopiowania pamięci EPROMs  
Roczna gwarancja.

(SB — 29)

## ATARI SPECTRUM

- programy użytkowe, edukacyjne, gry
- informacje po nadstaniu koperty zwrotnej ze znaczkiem

„P.K.T.S.” STUDIO KOMPUTEROWE  
WARSZAWA

ul. Królewska 43 m 25.

(SB 22)

**Sprzedaż detaliczną mode-  
mów ATARI-XM 301P oraz  
innych akcesoriów prowa-  
dzi sklep „HOBBY” ul.  
Sienna 89, 00-815 Warsza-  
wa.**

(SB-21)

Wymienię programy dla SPEC-  
TRUM, ATARI, COMMODORE.  
Janusz Wałaszek, skrytka pocztowa 1, 33-106 Tarnów 8.

(SB 25)

#### ATARI XL/XE

gry, programy użytkowe, edukacyjne.  
Najniższe ceny, najkrótsze terminy.  
11-035 Unieszewo 85 D/13

(SB 23)

Gry, nieśmiertelność, opisy, programy użytkowe, najtaniej oferuje „ATARI-BABY”. 42-640 Piekary Śl. Damrota 6a, tel. 87-37-70.

(SB 24)

Sprzedam drukarkę COMMODORE MPS1200. Oferty proszę kierować pod adres: Wojciech Barowski, Pl. Wojska Polskiego 2a/3, 58-301 Wałbrzych.

(SB 27)

**MÓZGOPROCESOR!** — rewelacyjną polską grę przygodową dla ZX Spectrum (program + kaseeta + opis 3100 zł) otrzymasz pisząc: COMPUTER ADVENTURE STUDIO, Bochnia 32-700, ul. Kazimierza Wielkiego 37/45.

(SB 20)

#### ZX SPECTRUM

Naprawy komputerów i klawiatur dla instytucji woj. katowickiego — rachunki.  
Zakład naprawy urządzeń elektronicznych 41-813 Zabrze-Helenka, ul. Jaworowa 6/2  
Czynne w godz. 18.00—20.00

(SB26)





## PLOTERY TABLICOWE

Typ	Format	Cena US\$
DMP-29M	A4/3	2570

## PLOTERY BĘBNOWE

E595DN PC	A4	720
E795DN PC	A4/3	895
EDMP-40	A4/3	1338
EDMP-52	A2/1	3695
EDMP-52MP	A2/1	4479
EDMP-56B	A4/3/2/1/0	6160
EDMP-61	A4/3/2/1	4825
EDMP-62	A4/3/2/1/0	6490

## PLOTERY TNĄCE DO FOLII

Typ	Szer. nośnika	Cena DM
DMP-40-V	400-420 mm	11.000



## PISAKI

Typ	Kolor	Cena US\$
Water based	8 kolorów	2,95/1 szt.

Kolory: czerwony, niebieski, fioletowy, brązowy, pomarańczowy, czarny, żółty



## AKCESORIA DIGITIZERÓW SERII HIPAD PLUS

Typ	Cena US\$
(dla modeli 90..)	
Sty90	2-button tilt correction stylus 33,55
CSR90-4	4-button cursor 55,95
(do modeli 92..)	
393-510	16-button cursor 331,50
393-507	16-button plexi cursor 443,50
393-508	16-button lighted cursor 667,50
393-509	stylus 156,75
393-400	power pack 220 V 70,00
xxxx	25 pin RS-232-C interface cabel 30,00

## DIGITIZERY SERII HIPAD PLUS

Typ	Format	Cena US\$
9012	305x305 mm	555
9018	305x457 mm	890
9236	610x914 mm	4250
9248	914x1219 mm	5370
9260	1118x1524 mm	6045

Zainteresowanych innymi produktami prosimy o bezpośredni kontakt



# ABC Data®

Ditmar-Koel-Str. 13  
2000 Hamburg 11, RFN  
tel. 040 314003/3195874  
tlx. 2166 002 abch d  
fax. 040 3191783

## ZAMÓWIENIE

Niniejszym zamawiam następujące artykuły:

..... sztuk.....	DM.....
..... sztuk.....	DM.....
..... sztuk.....	DM.....
..... sztuk.....	DM.....
Transport (1 szt. DM 40	
Kwota pobierana przez bank	DM 10,-
Razem:	DM.....

W załączeniu kopia zlecenia bankowego na przelew w/w sumy na konto ABC Data GmbH w Dresdner Bank, Hamburg, RFN, BLZ: 200 800 00, konto nr: 061 612 61.

W/w sprzęt proszę wysłać na adres:

Nazwisko i imię .....  
Kod pocztowy i miejscowość .....  
Ulica i numer domu ..... tel. ....  
Nazwisko, imię i adres zamawiającego (jeśli inne niż odbiorcy): .....

data ..... podpis .....

## JAK ZAMAWIAĆ ARTYKUŁY OFEROWANE PRZEZ ABC DATA?

- Po dokonaniu przelewu prosimy o wysłanie do nas załączonego zamówienia lub krótkiego listu (najlepiej na odwrocie kserokopii dowodu wpłaty) z dokładną informacją, co Państwo zamawiają i na jaki adres towar ma być wysłany.
- Wyroby nasze mogą Państwo również kupić osobiście w Berlinie Zachodnim lub zamówić wysyłkę z Bonn:  
ABC Computersystems GmbH  
Alt-Moabit 80  
1000 Berlin 21  
West Berlin  
tel. (030) 3915090/99  
tlx. 18 13 65 abc d  
fax. (0049-30) 3936483  
Bank für Handel und Industrie  
BLZ 100 800 00  
Konto nr 06 358 810 00  
Konto dol. 06 358 810 00/400  
3. Na wszystkie zakupione u nas produkty udzielamy 1-letniej gwarancji z obsługą serwisową prowadzoną w Polsce przez zakłady państwowe i prywatne.  
ABC Data GmbH jest firmą zarejestrowaną w Amtsgericht Hamburg, HRB 38590.



Impreza, na której winienes być obecny

# INFORMACJA '89

10 - 14 październik '89

Hala Widowiskowo-Sportowa "Spodek"-Katowice

- Informatyka w zarządzaniu - INFO '89
- Informatyka w medycynie - INFOMED
- Międzynarodowe sesje z udziałem  
Stowarzyszenia Dziennikarzy Nauki  
i Techniki
- Seminaria o tematyce: informatyka  
audio-video  
telewizja satelitarna

Jeśli chcesz być obecny wytnij poniższy kupon i wyślij

INF '89

Jestem zainteresowany:.....stoisko pokazowe  
.....stoisko informacyjne

Nazwisko.....

Stanowisko.....

Firma.....

Dziedzina zainteresowań.....

Adres.....

.....

telefon..... telex.....

Adresat:

Janusz Gołuch

"PRO - INFO"

40-001 Katowice 1 skr. poczt. 1347

tel. (832) 53-42-88

tlx. 312401 info pl







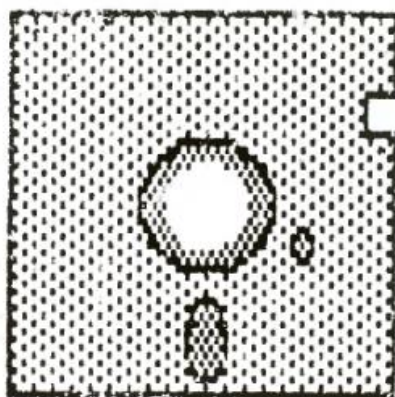
	GIEŁDA „BAJTKA” (tys.zł)	PEWEX BALTONA (USD)	RFN (śred.) (DM)
<b>SINCLAIR</b>			
ZX-Spectrum 48 KB	350	—	90-150
ZX-Spectrum plus	370	—	99-160
ZX-Spectrum + 2	500	—	180-280
ZX Spectrum - 3	850	—	280-400
TIMEX 2048	370	—	—

<b>COMMODORE</b>			
C-64C	650	219	275-350
C-128	850	299	399-500
C-128 D	1.1	—	777-800
Stacja dysków 1541	650	—	315-350
Stacja dysków 1570	—	—	369-400
Stacja dysków 1571	700	—	495-500
Drukarka MPS 804	450	—	—
Drukarka MPS 803	500	—	250-275
AMIGA 500	2.5	—	888
C-16	250	—	90-120
C-116	200	—	80-120

<b>ATARI</b>			
65 XE	480	114	100-180
130 XE	550	186	299
XC 12	80	36	50-70
Stacja dysków 1050	550	—	299
LDW 200	600	186	—

<b>AMSTRAD</b>			
464 mono monitor	750	—	320-450
464 kolor monit.	850	—	450-500
6128 mono monit.	1.3	—	680-850
6128 kolor monit.	1.5	—	770-900
PC 1512 SD MM	—	—	900-1000
Dyskietki 5.25	1.2-4.5	—	0.6-4
Dyskietki 3.50	3.5-5.0	—	3-9
Dyskietki 3.0	6.5-8.5	—	4-9
Joystick	12-45	—	6-25

<b>Sklep BAJTKA w Bytomiu</b> ul. Koniewa 6 tel. 81-57-01	
ZX-Spectrum 48 K	130.000
ZX-Spectrum plus	175.000
ZX-Spectrum +2	280.000
SEIKOSHA GP-50S	120.000
Commodore 64	240-270.000
Commodore 128	380.000
Commodore 128 D	—
AMIGA 500	—
Commodore 16	120.000
Commodore 116	100.000
Commodore +4	165.000
Magnetofon 1530	50-60.000
Stacja dysków 1541	270.000
ATARI 65 XE	190.000
ATARI 130 XE	260-290.000
XC-12	60-65.000
Stacja dysków 1050	240-270.000
LDW 2000 Super	270.000
464 mono	340.000



## INDYWIDUALNY BANK DANYCH

**Sebastian Siedlecki lat 16.** Posiada komputer IBM PG/AT kompatybilny z IBM PC/XT, kolorowy monitor oraz podwójną stację dysków. Oprogramowanie: około 100 gier oraz programy użytkowe. Nawiąże kontakt z posiadaczami IBM. Adres: 12 GOLDFINCH CRT 412, WILLOWPALE — ONTARIO, M2R 2C4, CANADA.

**Andrzej Samojednik, uczeń lat 19.** Posiada mikrokomputer ATARI 800 XL z magnetofonem i drukarkę Epsion. Oprogramowanie: ok. 150 programów, głównie użytkowych. Proponuje wymianę oprogramowania, literatury, oraz doświadczeń. Adres: 37-620 Horyniec Zdrój, ul. Mickiewicza 3.

**Adam Kołaczowski, uczeń lat 15.** Posiada COMMODORE 116 z magnetofonem. Oprogramowanie: około 150 gier oraz programy użytkowe. Za interesowania: technika video oraz informatyka. Proponuje wymianę gier, literatury. Adres: 02-373 Warszawa, ul. Mątwicka 8a m 2.

**W celu wymiany doświadczeń i oprogramowania** nawiąza kontakt dwaj posiadacze mikrokomputera "ENTER PKISE 128 ze Lwowa. Korespondencja w języku polskim lub niemieckim. **Adresy: Zawadka Aleksander (26 lat) ZSRR, 290054 Lwów, Prospekt Pobiedy 15/124. Martinek Georgij (27 lat) ZSRR, 290005 Lwów, ul. Draganinowa 56 m 7.**

**Władysław Babuska, lat 16.** Nawiąże kontakt z posiadaczami mikrokomputera Timex w celu wymiany doświadczeń oraz oprogramowania. Adres: PRAHA 7 — 17095, ul. Stupkowa 4, CSSR. Korespondencja w języku polskim.

**Grzegorz Zakolski, lat 16.** Posiada ATARI 800 XL, stację dysków LDW 2000 oraz ponad 400 gier i programów użytkowych. Nawiąże kontakt z posiadaczami ATARI w celu wymiany literatury i oprogramowania na dyskach. Adres: 06-100 Pułtusk, ul. Armii Czerwonej 5.

**Wojciech Mościbrodzki, uczeń LO.** Posiada ATARI 520 STFM z peryferiami. Nawiąże korespondencję z posiadaczami ST w celu wymiany oprogramowania, doświadczeń i literatury. Adres: 80-452 Gdańsk, ul. Kilińskiego 23 m2.

**Mariusz Kitzol — student, 22 lata.** Posiada ZX SPECTRUM oraz około 1000 programów i gier. Proponuje wymianę oprogramowania. Adres: 53-505 Wrocław, ul. Lelewela 6/7.

**Robert Stuczyński, uczeń 15 lat.** Posiada mikrokomputer C-64C z magnetofonem 1530. Proponuje wymianę literatury i oprogramowania. Adres: 62-510 Konin, ul. Wyzwolenia 5/47.

**Krzysztof Kukla, lat 13.** Posiada COMMODORE PLUS 4, magnetofon 1531 oraz około 60 gier. Proponuje wymianę gier. Adres: 47-400 Racibórz, ul. Willowa 2.

**Adam Muzyka, lat 11.** Posiada ATARI 65 XE. Oprogramowanie około 80 gier oraz kilka programów graficznych. Proponuje wymianę gier. Adres: 50-361 Wrocław, ul. Piastowska 45/8.

**Uwaga! posiadacze MERITUM.** Sanatorium Rehabilitacyjne dla Dzieci, 32-052 Radziszów, woj. Kraków prosi posiadaczy wyżej wymienionego mikrokomputera o pomoc w uzyskaniu programów oraz informacji na temat MERITUM.

## SUPER OFERTA SUPER KATALOG ATARI XL/XE ZX SPECTRUM

- największy wybór opisów do gier i programów, literatura
- gry, programy narzędziowe, użytkowe, polskie programy edukacyjne
- programy na zamówienie dla rzeźmi
- usługowe wykonywanie wydruków uniwersalny system ATARI TURBO HAPPY WARP 1050 do samodzielnego montażu
- kupno-sprzedaż sprzętu komputerowego, interfejsów do magnetofonów zwykłych i TURBO, interfejsów CENTRONICS, joysticków, dyskietek i materiałów eksploatacyjnych

**SPRZEDAŻ WYSYŁKOWA**  
Na wszystkie usługi udzielamy gwarancji, wystawiamy rachunki  
**MEGA-BAJT**  
03-945 Warszawa  
ul. Paryska 17/29  
skr. poczt. 28  
tel. 17-76-16 D-18

## STUDIO KOMPUTEROWE „ARION”

oferuje:  
gry, programy użytkowe, literaturę na ATARI, SPECTRUM, COMMODORE +4,116.  
Dom Towarowy „Centrum”  
ul. Słowackiego 7  
58-300 Wałbrzych  
katalogi kopertą zwrotną K-47

**ATARI** — programy pocztą  
HOBBIT Sp z o.o.  
20-950 LUBLIN 1 SKR POCZT. 369  
oferuje szeroki wybór gier, programów, użytkowych, instrukcji, opisów.  
Szczegółowe informacje i katalog-gratis po załączeniu koperty i znaczka.  
Co piąty program bezpłatnie. D-11

Dyskietki firmowe PRECISION, BASF, 3M, MAXELL, BONUS, DYSAN, VERBATIM — najtaniej od dostawcy z U.S.A.  
Informacje, cenniki: „Elektronika”, Kraków, Proszowicka 9, tel. 34-19-10. D-207

## INTERSOFT

00-443 Warszawa ul. Górnośląska 9/11 tel 28-67-94 tix 81-72-45

## Programy Edukacyjne

	cenę	ZX SPECTRUM	AMSTRAD CPC	ATARI XL/XE
12. CYKLOTRON Zadania działania od podst., budowa oraz zastosowanie cyklotronu.	5,000	X		
13. HOOKE Prosta symulacja doświadczenia ilustrującego prawo Hooke’a.	4,000	X	X	X
14. POTENCJAŁY WĘZŁOWE Obliczenia obwodów rozgałęzionych prądu stałego metodą potencjałów węzłowych.	9,000		X	
15. WIRNAG Wizualizacja pola magnetycznego. Pakiet dwóch programów.	8,000		X	
16. APROKSYMACJA Wyznaczanie wielokątnego aproksymującego funkcje ciągłe - ilustracja graficzna.	4,000		X	
17. JEDNOKŁADNOŚĆ Program ułatwiający zrozumienie pojęć jednorodności i podobieństwa.	10,000		X	
18. PUNKT XYZ Program wyznaczający współrzędne przestrzenne widzenia układu współrzędnych.	8,000		X	
19. KLIMAT EUROPY Nauka wraz z ciekawymi faktami ze znajomości średnich temperatur i opadów w Europie.	7,000		X	
20. GRAMATYKA 2 Ćwiczenia z gramatyki rozbioru zdań.	7,000		X	
21. JĘZYK NIEMIECKI Nauka tłumaczenia zdań w oparciu o podręcznik L. Jankowskiego.	8,000		X	
22. BITWY Nauka kojarzenia faktów historycznych z wielkimi bitwami w historii Polski.	16,000		X	

JAK KUPIWAĆ PROGRAMY ? - indywidualnie: za gotówkę lub za zaliczeniem pocztowym - instytucjonalnie: za gotówką lub czekiem. Ceny programów nie zawierają ceny nośników. Na życzenie zamówienia realizujemy na nośnikach klientów. Nośniki: ZX Spectrum - tasma i dysk 3", Atari - tasma, Amstrad - dysk 3".

## INTERSOFT

SB 19

## TOMBAT WYPOŻYCZALNIA XL ATARI XE

- gry i programy użytkowe
- co piąty program gratis
- inne bonifikaty
- opisy gier i instrukcje
- pomoc dla początkujących
- wysyłka na cały kraj
- katalog gratis

Nasz adres:  
ul. Magistracka 27 m 26  
01-413 Warszawa  
Tel. 363-078 godz. 12-20  
Zapraszamy!

(SB 12)



# Drogi Bajtku!

Na listy czytelników odpowiada  
**DOMINIK FALKOWSKI**

Moje pytania dotyczą komputera C-128D.

1. Czy stację dysków VIC-1541 można podłączyć do komputera C-128D w celu np. skopiowania programu w trybie 64?
2. W czasie współpracy C-128D z Neptunem 156 czy będzie możliwe uzyskanie 80-cio znakowego wiersza?

(nazwisko i adres do wiadomości Redakcji)

1. Komputery C-128 i C-128D w trybie pracy C-64 akceptują wszystkie peryferia jakie były dotąd stosowane przez C-64. Oznacza to, że po przejściu w tryb pracy C-64 wbudowana stacja 1571 automatycznie zamienia się na wolniejszą stację 1541. Podobnie sprawa się ma z komputerem C-128 z dołączoną zewnętrzną stacją 1571 lub 1570. Podłączanie dodatkowych stacji np. 1541 jest możliwe i to nie tylko do pracy w trybie C-64, ale również w trybach C-128 i CP/M. W tych dwóch ostatnich trybach stacja 1541 jest nieznacznie wolna i traci się całkowicie przyjemność pracy z tak wartościowym oprogramowaniem jak dBase, Turbo Pascal, Multiplan itp. Stacja bowiem powinna być, ale nie powinno się odczuwać jej obecności.

2. Wiele osób podłącza Neptuna 156 do komputerów C-128, a również do IBM i kompatybilnych, i z tej współpracy są zadowolone twierdząc, że uzyskiwana jakość obrazu jest wysmienita. Ja tę jakość uznaję za wystarczającą, jeżeli nie ma nic lepszego pod ręką (jest to całkowicie subiektywne odczucie). Osobiście uważam, że ze wszystkich urządzeń podłączanych do komputera monitor powinna cechować najwyższą jakość, gdyż w grę wchodzi nasze oczy. W związku z tym uważam, że warto zainteresować się monitorem z prawdziwego zdarzenia firm zachodnich. Praca z takim monitorem jest nieporównywalnie przyjemniejsza.

Posiadam komputer ATARI 800XL. Interesuję się programowaniem w kodzie maszynowym. Proszę

o odpowiedź na pytanie jak rozkazem BNE wykonać skok o np. 4 adresy do przodu.

Jarosław Kucuz  
ul. Żylewicza 5/60  
80-287 Gdańsk-Morena.

Procesory 6502, jak i inne z tej rodziny oprócz instrukcji rozgałęzienia BNE posiadają całą serię instrukcji tego typu. Są to BCC, BCS, BEQ, BMI, BPL, BVC, BVS. Każda z tych instrukcji po spełnieniu określonego warunku powoduje przerwanie sekwencyjnie wykonywanego programu i przekazuje wykonywanie programu do instrukcji innej niż ta bezpośrednio po instrukcji typu Bxx. I tak np. BNE \$DF25 w wypadku gdy znacznik Z w rejestrze słowa stanu ustawiony jest na zero (dla wyjaśnienia tych pojęć patrz Klan Commodore, cykl wykładów o monitorach poczynając od numeru 12/87) spowoduje wykonanie programu od adresu \$DF25, w przeciwnym razie wykonana będzie kolejna instrukcja po BNE. Tak więc chcąc dokonać skoku do dowolnej komórki należy podać jej adres w instrukcji Bxx.

Co jednak oznaczają poszczególne instrukcje:  
BCC adres — rozgałęzienie, gdy znacznik C(przeniesienia) ustawiony jest na 0  
BCC adres — jak wyżej z tym, że znacznik ustawiony jest na 1  
BEQ adres — rozgałęzienie, gdy wynik równa się zero  
BNE adres — rozgałęzienie, gdy wynik jest różny od zera  
BMI adres — rozgałęzienie, gdy wynik jest ujemny (znacznik ujemności ustawiony na 1)  
BPL adres — rozgałęzienie, gdy wynik jest dodatni (znacznik ujemności ustawiony na 0)  
BVC adres — rozgałęzienie, gdy znacznik V(przepięcia) ustawiony jest na 0  
BVS adres — rozgałęzienie, gdy znacznik V ustawiony jest na 1

Od dawna zafascynowany jestem Turbo Pascalem 3.0 i w mojej pracy na IBM PC/AT staram się korzystać wyłącznie z tego języka. W trakcie pracy nasunęło mi się jedno pytanie. Jaka instrukcja w Turbo-Pascalu drukuje program na drukarce.  
(nazwisko i adres do wiadomości Redakcji)

Bezpośredniej instrukcji w Turbo-Pascalu nie ma. W celu wydrukowania zbioru proponuję posłużyć się niżej przedstawionym programem.

```
program druk;
var
  Textfile:Text;
  Scratch:String[128];
begin
  write('Podaj nazwę zbioru');
  readln(Scratch);
  assign(Textfile,Scratch);           {otwórz zbiór}
  {$I-}
  reset(Textfile);
  {$I+}
  if IO result = 0 then
    writeln('Zbiór',Scratch,' nie został znaleziony')
  else
    begin
      while not eof(Textfile) do
      begin
        readln(Textfile,Scratch);
        writeln(Lst,Scratch);
      end;
      writeln(Lst)                    {opróżnia bufor drukarki}
    end
  end.
```

„Jakie języki programowania są dostępne dla komputerów Commodore (interesują mnie przede wszystkim C-64 i C-128)?”

Commodore 64 i 128 doczekały się bardzo wielu rozszerzeń oryginalnego interpretera BASIC oraz licznych



Adres: Przedsiębiorstwo "PRO-INFO"

ul. Sikorskiego 18/38

40-001 KATOWICE

skrytka pocztowa 1347

tel. 53-42-88

## OFERTA

### Oprogramowanie oraz dokumentacja

IBM, Atari 800/65 XE/ST, Amstrad, Commodore, Amiga

IBM, Clipper86/87 wersja polska, Turbo Pascal v4.0, Turbo C,  
Turbo Basic, dBase III+, Pro-Desin, DOS-3.3, Xenix

### oraz organizacja imprez promocyjnych

Masz pomysł - napisz

Masz dokumentację w wersji polskiej - napisz

Masz ciekawy program - napisz



interpreterów bądź kompilatorów innych języków. Do najważniejszych należą:

a) grupa rozszerzeń BASIC:

EXBASIC LEVEL II, SIMON'S BASIC, METABASIC, OMICRON BASIC, GRAPHIC BASIC, ULTRA BASIC, ACCESS BASIC, BASIC V4.0, BASIC V4.2, WARSAW BASIC;

b) kompilatory/interprety:

USCO PASCAL, OXFORD PASCAL (ten ostatni ma kilka różnych wersji również taśmowych), PROFI PASCAL, F.I.G. FORTH, FORTH DATATRONIC, SUPER-FORTH, LISP, VANILLA PILOT, PROLOG, PROMAL, COMAL, LOGO, FORTRAN 64, C, SUPER-C, COBOL. W wypadku C-128 oprócz wymienionych powyżej języków programowania oferta rozszerza się o języki dostępne pod kontrolą systemu CP/M, a więc TURBO-PASCAL V2.0 i V3.0, NEVADA FORTRAN, BASIC 80, PL/I, MICROPROLOG, FORTRAN 80, COBOL itp. w samym trybie pracy C-128 dostępny jest również kompilator języka C.

**„Moje Atari 65 XE zostało zalane niechcący wodą sodową i natychmiast odmówiło posłuszeństwa... [...] Gdzie mogę naprawić mój komputer?”**

**„[...] Udało mi się kupić wyjątkowo tanio C-16 na giełdzie jak się potem okazało głównie dlatego, że był on zepsuty. Redakcja pomóż mi znaleźć warsztat gdzie mógłbym naprawić C-16 [...]”**

**„Wystarczyło jedno zahaczenie o przewód zasilania i moje Spectrum „padło”. Czy jest taki punkt w Warszawie gdzie można by go naprawić [...]”**

Takich listów dostajemy coraz więcej i dlatego też postaramy się pomóc naszym Czytelnikom. A oto adresy zakładów naprawiających komputery:

COMMODORE:

1. Warszawa, ul. Instalatorów 9 (tylko serwis gwarancyjny komputerów zakupionych w przedsiębiorstwie „PEWEX”!),

2. Firma Andrzeja Raczkiewicza, Warszawa, ul. Andersena 3 m. 103,

ATARI:

1. PZ „KAREN”, Warszawa ul. Obrońców 23, tel. 17-84-10

ZX SPECTRUM:

1. Warsztat prywatny, Warszawa, ul. Mokotowska 39. Listę tą będę sukcesywnie uzupełniał o dalsze adresy o ile otrzymam odnośne informacje od Czytelników lub zainteresowanych placówek.

**„Czy możliwe jest rozszerzenie pamięci komputera 8-bitowego w taki sposób abym mógł mieć do dyspozycji 640 KB RAM?”**

I tak i nie. Samo rozszerzenie pamięci jest oczywiście możliwe z tym, że nie pozwala ono na rozszerzenie np. samej pamięci operacyjnej do 640 KB. Mikroprocesor 8-bitowy pracujący zwykle w oparciu o 16-bitową szynę adresową jest w stanie zaadresować jednocześnie maksymalnie 65536 komórek pamięci ( $2 \uparrow 16 = 65536$ ). Wynika więc z tego, że bez względu na to ile pamięci dodamy do naszego komputera to nadal podstawowym obszarem adresowym będzie 64 KB RAM.

Z drugiej strony są w sprzedaży takie moduły — np. do komputera C-64 opracowano kartę zawierającą 1 MB RAM pamięci dodatkowej. Jej wykorzystanie jest jednak ograniczone — można w tej pamięci zapisać np. szereg obrazów graficznych wykorzystując ją przy tym jako RAM-DISK lecz nie daje to ani bajta więcej dla interpretera BASIC. Winę za to ponosi jednak przede wszystkim procesor.

**„Jaka jest różnica pomiędzy przerwaniem IRQ i NMI? Do czego właściwie służą przerwy w komputerze?”**

Przerwy typu IRQ (Interrupt ReQuest) są przerwami maskowalnymi tzn. takimi, które w sposób programowy można wyłączyć (np. za pomocą rozkazu SEI). Z kolei przerwy typu NMI (Non Maskable Interrupt) wyłączyć już nie można, gdyż mają one najwyższy możliwy priorytet. To jest główna różnica pomiędzy obydwoma typami przerw.

Przerwy w komputerze służą do wykonywania wielu operacji niezbędnych dla funkcjonowania samego komputera. Najlepszym przykładem takiej procedury może być np. testowanie klawiatury, czy nie został wciśnięty jakiś klawisz. Jeśli tak się stało, to komputer automatycznie podejmuje dalsze działanie mające na celu zasygnalizowanie błędu, wykonanie danej operacji, wpisanie linii programu do pamięci czy wyświetlenie danego znaku na ekranie. Aby taki test i działanie było możliwe, komputer musi sprawdzać bardzo często, czy operator nie wcisnął jakiegoś klawisza; zwykle test taki jest wykonywany 50 razy na sekundę. Oczywiście jest to z konieczności najprostszy przykład, gdyż dokładne omówienie tematu znacznie wykracza poza łamy tej rubryki.

**1. W żadnym z numerów „Bajtki” nie znalazłem informacji na temat różnych wersji komputerów ST. Co kryją skróty STF, STM i STFM?**

**2. Czy w komputerach Atari ST można uzyskać tryb wysokiej rozdzielczości na monitorze kolorowym?**

**3. Czy na Zachodzie zabronione jest kopiowanie programów wyłącznie na własny użytek?**

**4. Czy powstał już emulator IBM na Atari ST?**

**Piotr Lenarczyk  
Rzeszów**

1. Dodatkowe litery w nazwie komputerów ST są symbolami wbudowanych do komputera dodatkowych urządzeń. Litera M oznacza modulator TV, który umożliwia wykorzystanie zwykłego telewizora kolorowego jako monitora, natomiast litera F oznacza stację dysków 3,5 cala.

2. Z trzech trybów pracy monitora ST (niska, średnia i wysoka rozdzielczość) dwa pierwsze są przeznaczone wyłącznie do pracy z monitorem kolorowym, a trzeci wyłącznie z monitorem monochromatycznym. Zamiast monitora kolorowego do pracy z komputerami wyposażonymi w modulator można wykorzystać odbiornik telewizyjny. Rodzaj monitora jest rozpoznawany przez komputer automatycznie i nie można go „zmusić” do pracy w niedozwolonej dla danego monitora rozdzielczości. Jedynym sposobem uzyskania na monitorze kolorowym trybu wysokiej rozdzielczości jest zastosowanie specjalnego programu o nazwie „Monoware”. Efekt takiej operacji jest jednak mało interesujący.

3. W krajach rozwiniętych programy komputerowe są chronione na równi z każdym innym rodzajem twórczości. Zabronione jest więc także ich powielanie w dowolnej formie. Jedynym wyjątkiem jest możliwość wykonania zapasowej kopii posiadanego legalnie (zakupionego) programu wyłącznie dla własnych potrzeb. Odstępowanie tej kopii innym osobom jest zabronione (nawet nieodpłatnie).

4. Na komputery ST powstał doskonały emulator IBM o nazwie „PC Ditto”. Ma on dwie wersje przeznaczone dla różnych monitorów. Pracuje z każdym modelem ST, lecz pełna zgodność z IBM wymaga 1 MB pamięci RAM. Na 520ST emulator nie daje pełnej zgodności z IBM, gdyż pozostały obszar pamięci jest mniejszy niż w najmniejszej konfiguracji IBM. W takiej sytuacji niektóre duże programy nie mogą pracować.

**1. Czy uwagi zawarte w artykule „Nie tylko dla nerwowych graczy” (Klan Commodore 4/88) mogą dotyczyć Atari 800XL?**

**2. Czy można zamontować w magnetofonie wzmacniacz sygnału podawanego na głowicę podczas nagrywania z komputera?**

**3. Czy można podłączyć komputer Atari 800XL do magnetowidu tak, aby nagrywać obraz i dźwięk?**

**4. Jaka instrukcja obsługuje pióro świetlne?**  
**Aleksander Juszyński  
Lubaczów**

1. Oczywiście wszystkie umieszczone tam uwagi dotyczą także Atari XL/XE. Jedynym wyjątkiem jest ko-

mentarz do ostatniej uwagi. W Atari joysticki są obsługiwane przez inny układ niż klawiatura i dlatego nic na ekranie się nie ukaże.

2. Można, ale po co? Moc sygnału nagrywającego jest tak dobrana, aby umożliwić najlepszy zapis. Jej zwiększenie może spowodować przesterowanie zapisu i błędy przy późniejszym odczycie. Jeśli występują kłopoty z zapisem, to należy oddać magnetofon do regulacji w punkcie serwisowym.

3. Komputery XL/XE można połączyć z magnetowidem dwoma sposobami: przy pomocy przewodu antenowego z gniazda TV komputera do gniazda antenowego magnetowidu lub ekranowanym przewodem dwużyłowym z gniazda MONITOR w komputerze do gniazda AUDIO-IN i VIDEO-IN w magnetowidzie. Wyprowadzenie sygnałów AUDIO i VIDEO na styki w gnieździe MONITOR było pokazane w drugim „Bajtku — Tylko o Atari”. Do magnetowidu będzie przesyłany dokładnie taki sam obraz i dźwięk, jak do monitora, przy czym drugi sposób daje lepszą jakość przekazu.

4. Nie ma specjalnych instrukcji do obsługi pióra świetlnego. Jego położenie można odczytać przy pomocy instrukcji PEEK z komórek 564 (poziomo) i 565 (pionowo). Położenie jest w nich określane z dokładnością do jednej linii ekranu (nie obrazu) w pionie i dwóch punktów ekranu w poziomie.

**Czy możecie napisać coś o przerwaniach w Spectrum? Na temat przerw w innych komputerach jest coś w każdym prawie numerze Bajtki, tylko Spectrum jest poszkodowane. Podobno przerwy jest kilka rodzajów, ale czym się różnią i czy można z nich korzystać — tego nie wiem.**

**M. Bąk, Warszawa**

Rzeczywiście, mikroprocesor Z80 dysponuje kilkoma trybami przerw, jednak w Spectrum ich wykorzystanie napotyka na spore trudności. Przerwy programowe, mimo ich istnienia, są mało użyteczne, gdyż w większości są wykorzystywane przez interpreter BASIC-a do własnych, specyficznych celów, toteż z punktu widzenia użytkownika Spectrum do wykorzystania nadają się tylko przerwy zewnętrzne polegające na tym, że na sygnał z zewnątrz (generowany w Spectrum 50 razy na sekundę) mikroprocesor porzuca program i zaczyna wykonywać odpowiedni podprogram. W pierwszym trybie przerw jest to podprogram zaczynający się pod adresem #0038. Jest on zapisany w ROM-ie, obsługuje klawiaturę i zwiększa o jeden zmienną systemową FRAMES. W drugim trybie adres procedury obsługującej przerwy musi być zapisany w odpowiednim miejscu pamięci. Po przyjęciu sygnału przerwy mikroprocesor konstruuje adres pod którym będzie szukać adresu procedury obsługi przerwy z dwóch bajtów — starszy to zawartość rejestru I, młodszy jest odczytywany z magistrali danych. Dzięki temu różne urządzenia zewnętrzne mogą wywoływać „swoje” procedury obsługi. Niestety, konstruktorzy Spectrum nie przewidywali możliwości zastosowania drugiego trybu przerw w programach, toteż ich użycie wymaga zastosowania odpowiednich trików, i bez ingerowania w sprzęt pozwala tylko na zastąpienie procedury spod adresu #0038 inną. Postaramy się zająć tym problemem na naszych łamach.

Oprócz wymienionych istnieją jeszcze inne rodzaje przerw, których ze względu na konstrukcję Spectrum nie daje się wykorzystać. Radzę poszukać informacji na ich temat w książce o mikroprocesorze Z80.

**Jak to jest z uruchamianiem na Spectrum programów w kodzie maszynowym? Nigdzie nie znalazłem informacji o tym, że można to zrobić inaczej, niż przy pomocy instrukcji RANDOMIZE USR, tymczasem niektóre gry wczytuje się przy pomocy LOAD ""CODE, i uruchamiają się same. Czy to znaczy, że istnieje jeszcze jakaś możliwość wywołania kodu maszynowego?**

**Marek Borzuchowski, Kielce**

Niestety, w Spectrum wywoływać programy napisane w kodzie maszynowym można tylko przy pomocy funkcji USR (niekoniecznie jako RANDOMIZE USR, może to być również LET a=USR adres, lub PRINT USR adres). Programy uruchamiane w opisany przez Pana sposób, również korzystają z funkcji USR, o ile są napisane w kodzie maszynowym. Wczytywany blok bajtów zawiera obszar zmiennych systemowych i programu napisanego w BASIC-u. Interpreter po wczytaniu z taśmy dowolnej informacji zaczyna działać zgodnie z tym, co jest zapisane w zmiennych systemowych. Jeżeli wynika z nich, że powinien wykonać konkretną instrukcję z konkretnej linii robi to. Radzę zrobić następujący eksperyment: wpisać dowolny program w BASIC-u, dodać do niego linie

1 SAVE "Przykład" CODE 23552,256\*PEEK 23642+PEEK 23641-23522

uruchomić i nagrać kod na taśmę (adres i długość bloku kodu są tak dobrane przy pomocy zmiennych systemowych, by na taśmę zostało nagrane wszystko to co jest potrzebne). Potem po wyzerowaniu komputera wystarczy napisać LOAD ""CODE i uruchomić magnetofon. Myślę że po tej próbie wszystko będzie jasne.

Teoretycznie można się pokusić o uruchomienie programu korzystając ze zmiennej systemowej ERR SP, informującej gdzie szukać adresu procedury obsługującej błędy. Byłby to jednak sposób bardzo trudny, wymagający sporej wiedzy na temat Spectrum i organizacji jego interpretera. Takie chwytły mają sens tylko wtedy, gdy staramy się zabezpieczyć program przed analizą niepowołanych osób.

(SB 18)

## JOYSTICK SERVICE

Zgłoszenia: Studio komputerowe SEZAM  
D.H. "SEZAM" II p. — czwartki 16" — 19"

Korespondencja: JOYSTICK SERVICE

02-770 Warszawa 130 skr. poczt. 102  
tel. grzechn. 41-22-22

D 241



# KOMPUTER O MÓZDZKU PSZCZOŁY

„Jeśli się dobrze przyjrzeć informatyce, to właściwie od jej powstania 45 lat temu, nie wydarzyło się nic nowego” stwierdza Daniel Garric w styczniowym numerze *Le Point*. Komputery są co prawda coraz szybsze, mniejsze i łatwiejsze w obsłudze, ale dopiero dzisiaj stoimy na progu nowej rewolucji informatycznej — stworzenia sztucznej inteligencji.

Tym razem celem nie jest zbudowanie jeszcze szybszego komputera, którego tam generacji. Naukowcy mierzą wyżej. Chodzi o skonstruowanie maszyny działającej na zasadach zbliżonych do biologicznego mózgu.

Od trzech lat najbliższym przyjacielem człowieka w Bell Laboratories, tych samych w których 40 lat temu na zamówienie wynaleziono tranzystor, jest ślimak. Jego największą zaletą jest mózg — łatwo dostępny, nieskomplikowany, o dużych komórkach nerwowych i neuronach. Zadaniem naukowców jest zrozumienie istoty jego działania i odwzorowania jej na komputerze.

W USA nad tym zagadnieniem pracują cywilne i zwłaszcza wojskowe ośrodki badawcze dysponujące budżetami sięgającymi dziesiątków milionów dolarów. W Japonii przy zapewnieniu niezbędnej ilości jenów i specjalistów, realizowany jest międzynarodowy program „Ludzkie gra-

nice”. W Europie utworzono projekt „Brain” (mózg), w którym uczestniczą wszystkie najbardziej znaczące firmy elektroniczne.

Na czym ma polegać postęp? Weźmy prosty przykład. Jeśli do pamięci komputera włożymy milion fotografii osób, po czym zaprezentujemy mu kopię jednej z nich i poprosimy o jej identyfikację, to on zrobi to w ciągu sekundy. Ale jeśli pokażemy mu inne zdjęcie danej osoby, nawet bardzo podobne, to odpowie, że tej osoby nie ma w swoim zbiorze danych. Jest po prostu głupi. Gdyż nawet bardzo ograniczony osobnik jest w stanie rozpoznać daną osobę na podstawie bardzo powierzchownej karykatury.

Rozwiązanie tkwi w przejściu od komputerów linearnych „kartezjańskich” do informatyki „neuralnej”.

Już twórca informatyki John von Neumann przeczuł, że przyszłość leży w neuronach. Niektórzy matematycy poszli tym tropem i ułożyli nawet parę

interesujących równań. Okazało się jednak, iż technologia i wiedza teoretyczna nie były wówczas wystarczająco zaawansowane. Dopiero w 1982 r. amerykański fizyk John Hopfield udowodnił, że istnieje pewna analogia między molekułami gazu w ruchu i zachowaniem „modelowych” neuronów (neuron modelowy jest przybliżonym odwzorowaniem neuronu biologicznego w sieci utworzonej z tranzystorów i mikroprocesorów).

Zważywszy, że znane są przecież prawa statystyki opisujące zachowanie gazów, był to wyłom, za którym poszła lawina. Od 1985 r. w największych laboratoriach na całej ziemi powstały międzydiscyplinarne zespoły badaczy grupujące neurobiologów, informatyków, matematyków, logików, fizyków i psychologów, kładące podwaliny pod nową dyscyplinę — „naukę o myśleniu”.

W ciągu trzech lat dokonano niebagatelnych postępów, czego przykładem może być Net-Talk — sieć neuronalna, która nauczyła się sama w ciągu 8 dni czytać słownik angielski zawierający 20 000 słów. Teraz potrafi przeczytać na głos dowolny tekst, stosując intonację wynikającą z interpunkcji i oddzielając wyraźnie słowa.

W najbliższych miesiącach na rynku USA pojawią się małe zastawy, potrafiące odczytywać napisane ręcznie czeki a nawet dłuższe informacje. A wszystko to na mikrokomputerach kompatybilnych z IBM czy Apple.

Najbardziej zainteresowana jest oczywiście armia. Radar potrafiący odróżnić samoloty własne od wrogich i neutralnych mógłby zaoszczędzić wiele istnień ludzkich unikając tragicznych pomyłek.

Każdy z nas ma w głowie ok. 100 miliardów neuronów. Jeden neuron ma przeciętnie ok. 1000 połączeń, co daje razem 100 000 miliardów połączeń. Z faktu, iż człowiek jest około dziesięć tysięcy razy wolniejszy od komputera wynika, że aby naśladować mózg ludzki komputer powinien posiadać 10 miliardów połączeń. Dziś maszyny posiadają do 20 000 połączeń, a za dziesięć lat dzięki dalszej miniaturyzacji, być może uda się osiągnąć 2 miliony. Na razie nie ma jednak nawet pomysłów, jak osiągnąć więcej. Ale na razie nikogo to nie martwi. Użyteczne będą maszyny o inteligencji dużo niższej niż ludzka. Specjaliści twierdzą, że do kierowania polem bitwy wystarczyłby im mózdek pszczoły.

Klasyczny komputer składa się z dwóch zasadniczych części: jednostki obliczeniowej, która dodaje, odejmuje i selekcjonuje dane oraz pamięci zawierającej programy i dane.

W mózgu biologicznym nie udało się do tej pory wyodrębnić pamięci ani nawet poznać jej istoty. Wiadomo tylko, że biologicznym ekwiwalentem jednostki centralnej komputera, jego programów i zbioru danych, jest system nerwowy.

Wydawałoby się, iż ta wiedza nie daje wielu wskazówek co do sposobu budowy maszyn. Jednak jest inaczej. Komputery nieliniarne budowane są z utworzonych na podstawie tranzystorów komórek, z których każda jest połączona z dwoma innymi komórkami. Każda z nich posiada własną pamięć. Całość tworzy sieć, po której wędrują impulsy odbierane z zewnątrz.

Jej działanie można wyobrazić sobie na podstawie funkcjonowania bilardu elektrycznego. Kule symbolizują impulsy elektryczne; przeszkody — neurony; licznik — pamięć. W zależności od tego na jaką przeszkodę trafi wypuszczona kula, licznik zapisuje odpowiednią ilość punktów. Niestety, rzeczywistość jest bardziej skomplikowana. Impulsy przychodzące z zewnątrz mają różną wartość w zależności od tego, jaką informację przenoszą, poza tym wchodzi ich do sieci wiele na raz. W bilardzie ilość przeszkód nie przekracza kilkunastu. W najbardziej zaawansowanych konstrukcjach sieciach, liczba neuronów osiąga tysiąc, a ponieważ każdy może być połączony z każdym, to otrzymujemy milion możliwości. Rozrysowanie schematu połączeń stanowi o ilości i rodzaju połączeń niezbędnych dla nadania sieci sensu i umożliwiających jej funkcjonowanie.

Na tym właśnie polega następna różnica między komputerem klasycznym a neuronalnym. Tego drugiego nie da się zaprogramować przekazując mu instrukcję działań krok po kroku. Wróćmy do przykładu z bilardem elektrycznym, który tym razem będzie przedstawiał sieć neuronalną. Założmy, że naszym celem jest osiągnięcie pewnej sumy punktów, która zapewni dodatkową darmową partię. Wypuszczamy jedną, drugą, dziesiątą kulę. Jeśli sieć jest dobrze zaprojektowana, to z początku komputer będzie się mylił, ale po pewnym czasie nauczy się tak dobierać kolejność i siłę uderzeń, żeby uzyskać oczekiwany rezultat.

Dla postronnego obserwatora zdumiewający może wydać się fakt, iż programiści sami nie wiedzą, w jaki sposób komputer się uczy. Powstaje natomiast nowa specjalizacja nauczycieli sieci neuronowych, którzy muszą się posługiwać w równej mierze wiedzą i intuicją. Zauważono na przykład, iż dla komputera nie jest obojętne, w jakiej kolejności podaje mu się przykłady. Po krótkim zastanowieniu nie jest to takie trudne do zrozumienia. Na mieszkańcu Krakowa gdańska starówka nie wywrze większego wrażenia. Natomiast mieszkaniec białostockiej wioski, który nigdy nie opuszczał województwa, będzie nią zachwycony. Ten proces nie poddaje się na razie żadnej kwantyfikacji, choć jest intuicyjnie zrozumiały.

A trudności właściwie dopiero się zaczynają. Wystarczy powiedzieć, że z badań nad neuronami biologicznymi wynika, iż o ile niektóre sygnały elektryczne są normalnie przekazywane między neuronami, to inne rozchodzą się w sposób przypadkowy, zamieniając się niekiedy na substancje chemiczne — nośniki informacji. Niektórzy naukowcy zastanawiają się, czy komórki nie grają w kości, próbując różnych sposobów, zanim znajdą dobry.

Pytań pozostaje więc bardzo wiele. Tym niemniej prace wdrożeniowe idą pełną parą i za 5 — 10 lat możemy się spodziewać pojawienia na rynku komputerów wielkości książki, z którymi komunikacja będzie możliwa za pomocą pisma odręcznego, nawet bez znajomości podstaw informatyki. Spora część ich umiejętności będzie polegała na poznaniu potrzeb właściciela i nauce ich zaspokajania. Co do daty dostarczenia wojskowemu mózgu pszczoły — źródła milczą.

opr. MR

